Proceedings of Machine Learning Research vol 242:1–12, 2024

Restless Bandit Problem with Rewards Generated by a Linear Gaussian Dynamical System

Jonathan Gornet JONATHAN.GORNET@WUSTL.EDU and **Bruno Sinopoli** BSINOPOLI@WUSTL.EDU
Department of Electrical and Systems Engineering, Washington University in Saint Louis

Abstract

Decision-making under uncertainty is a fundamental problem encountered frequently and can be formulated as a stochastic multi-armed bandit problem. In the problem, the learner interacts with an environment by choosing an action at each round, where a round is an instance of an interaction. In response, the environment reveals a reward, which is sampled from a stochastic process, to the learner. The goal of the learner is to maximize cumulative reward. In this work, we assume that the rewards are the inner product of an action vector and a state vector generated by a linear Gaussian dynamical system. To predict the reward for each action, we propose a method that takes a linear combination of previously observed rewards for predicting each action's next reward. We show that, regardless of the sequence of previous actions chosen, the reward sampled for any previously chosen action can be used for predicting another action's future reward, i.e. the reward sampled for action 1 at round $t-1$ can be used for predicting the reward for action 2 at round t. This is accomplished by designing a modified Kalman filter with a matrix representation that can be learned for reward prediction. Numerical evaluations are carried out on a set of linear Gaussian dynamical systems and are compared with 2 other well-known stochastic multi-armed bandit algorithms.

Keywords: Non-stationary stochastic multi-armed bandit, stochastic dynamical systems, Kalman filter

1. Introduction

The Stochastic Multi-Armed Bandit (SMAB) problem provides a rigorous framework for studying decision-making under uncertainty. The problem consists of the interaction between a learner and an environment for a set number of rounds. For each round, the learner chooses an action and in response the environment reveals a reward, which is sampled from a stochastic process, to the learner. The goal of the learner is to maximize cumulative reward. In the non-stationary case of the SMAB, the distributions of the reward for each action can change each round. A key result in the area is Besbes et al. (2014) where it assumes that the cumulative changes in the reward distributions are bounded by a known constant.

A more specific variation of the non-stationary SMAB are environments where the rewards are generated by s-step autoregressive models, i.e. an action's sampled reward X_t is a linear combination of rewards $X_{t-s}, \dots, X_{t-1}$ where s is the autoregressive model order. Two key results that have tackled this SMAB environment are Slivkins and Upfal (2008), Bogunovic et al. (2016), and Chen et al. (2023). Slivkins and Upfal (2008) studied the performance of a number of algorithms for rewards generated by Brownian motion. In Bogunovic et al. (2016), the authors consider when the rewards for each action is generated by a *known* 1-step autoregressive process. In Chen et al. (2023), they address SMAB environments modeled as an *unknown* 1-step autoregressive or a *known* s-step autoregressive. A key application of autoregressive models is presented in Parker-Holder et al.

(2020), where the work tunes the hyperparameters, such as the gradient descent's learning rate, during the training process of a reinforcement learning based on neural networks. Finally, another perspective to s-step autoregressive models is Gornet et al. (2022) where the reward X_t and a context θ_t are generated by a Linear Gaussian Dynamical System (LGDS), where a context is a partial observation of the LGDS's state variables. The authors prove that a linear combination of previously observed contexts $\theta_{t-s}, \dots, \theta_{t-1}$ can be used to predict the reward X_t, a perspective similar to the environments considered in Bogunovic et al. (2016) and Chen et al. (2023).

Our work proposes a discrete-time restless bandit with continuous state-space by assuming the state and rewards are generated by a LGDS. This paper extends the results in Gornet et al. (2022) where now the context is no longer observed. The contributions of our paper are as follows.

Our Contributions:

- We introduce a SMAB environment where the rewards are generated by a LGDS in Section 2.
- We prove that we can predict the reward for each action by using a linear combination of observed *rewards*. For example, for an environment with 3 actions, if a learner chose action $A_{t-2} = 2$ at round $t-2$ and $A_{t-1} = 1$ at round $t-1$, the learner can take a linear combination of the sampled rewards X_{t-2} and X_{t-1} to predict the reward for action $a = 3$ at round t. The coefficients for the linear combination are from the identified modified Kalman filter matrix representation. We provide a proof of the error bound of the reward prediction for the identified modified Kalman filter. The idea is inspired by Tsiamis and Pappas (2019) for identifying the Kalman filter, where now we assume that the measurements of the LGDS, a linear combination of the system's state variables, can change each round. (See Section 3)
- Using the proved error bound of the reward prediction, we propose the algorithm Uncertainty-Based System Search (UBSS). The algorithm chooses the action that maximizes the sum of the reward prediction and its error. (See Section 4)
- For numerical results in Section 5, we apply UBSS to a parameterized LGDS to illustrate its numerical performance. Here, we compare UBSS to Upper Confidence Bound (UCB) algorithm (Agrawal, 1995) and Sliding Window UCB (SW-UCB) (Garivier and Moulines, 2008) algorithm, two well-known SMAB algorithms, and for which LGDS UBSS performs best.

Note: For proofs of the lemmas and theorems, please refer to the ArXiv version found in Gornet and Sinopoli (2024).

Related Work

One example of the non-stationary SMAB is the restless bandit where the reward for each action is the function of a state that is generated by a Markov chain Whittle (1988). Whenever the learner chooses an action, the learner observes a Markov chain's state and a reward. This paper focuses on the case when the transition matrix of the Markov chain is unknown. Previous results in the discrete state-space Markov chain that use an approach similar to UCB are Tekin and Liu (2012); Ortner et al. (2012); Wang et al. (2020); Dai et al. (2011); Liu et al. (2011). Jung and Tewari (2019) uses Thompson sampling, i.e. sampling parameters based on *a priori* distribution of Markov chain, for action selection. We avoid comparisons with these previous results since the states of

the Markov chain are discrete, whereas the results presented in this paper focus on when the states are continuous. This allows us to tackle a different set of application, such as hyperparameter optimization for reinforcement learning based on neural networks, e.g. Parker-Holder et al. (2020).

2. Problem Formulation

The learner will interact with an environment modeled as a LGDS. We will consider the following LGDS:

$$\begin{cases} z_{t+1} &= \Gamma z_t + \xi_t, \;\; z_0 \sim \mathcal{N}\left(\hat{z}_0, P_0\right) \\ X_t &= \langle c_{A_t}, z_t \rangle + \eta_t \end{cases}, \tag{1}$$

where the reward $X_t \in \mathbb{R}$ is the inner product of an action vector $c_{A_t} \in \mathcal{A}$ and the state $z_t \in \mathbb{R}^d$. The process noise $\xi_t \in \mathbb{R}^d$ and measurement noise $\eta_t \in \mathbb{R}$ are independent normally distributed random variables, i.e. $\xi_t \sim \mathcal{N}(0, Q)$ and $\eta_t \sim \mathcal{N}(0, \sigma^2)$. The action vector $c_{A_t} \in \mathcal{A} = \{c_a \in \mathbb{R}^{d\times 1} \mid \|c_a\|_2 \leq B_c, a \in [k]\}$ where B_c is known and $a \in [k] \triangleq \{1, 2, \ldots, k\}$ is the indexed action. Using similar notation as Abbasi-Yadkori et al. (2011), actions that are realized at round t are denoted as $c_{A_t} \in \mathcal{A}$ and unrealized actions are denoted as $c_a \in \mathcal{A}$. We make the following assumptions on system (1).

Assumption 1 *The state matrix Γ is marginally stable, i.e. $\rho(\Gamma) \leq 1$.*

Assumption 2 *The vectors and matrices in system (1) are unknown along with Q, σ, and d. However, number of actions k is known.*

Assumption 3 *The matrix pair $\left(\Gamma, Q^{1/2}\right)$ is controllable. The pair $\left(\Gamma, c_a^\top\right)$ is detectable for every vector $c_a \in \mathcal{A}$.*

The goal of the learner is to maximize the cumulative reward over a horizon $n > 0$, i.e. $\sum_{t=1}^{n} X_t$. The horizon length $n > 0$ may be unknown. To provide analysis on the performance of any proposed algorithm for maximizing cumulative reward in (1), regret is analyzed which is defined to be

$$R_n \triangleq \sum_{t=1}^{n} \mathbb{E}\left[X_t^* - X_t\right], \tag{2}$$

where X_t^* is the highest possible reward that can be sampled at round t. In the next section, we discuss a reward predictor for the LGDS (1).

3. Predicting the Reward of the LGDS

This section reviews the optimal 1-step predictor of the rewards, in the mean-squared error sense, generated by LGDS (1): the Kalman filter. According to Assumption 2, the matrices of the LGDS (1) are unknown, implying that the Kalman filter needs to be identified. However, to the best of our knowledge, no current results exist for direct identification of the Kalman filter when the LGDS's (1) action vector $c_{A_t} \in \mathcal{A}$ can change each round. Therefore, we propose a modified Kalman filter to identify. Imposing the assumptions posed in the previous section, we prove that prediction error of the modified Kalman filter is lower than or equal to the variance of the reward X_t, making it

possible to extract a signal to predict the reward for each action. The added benefit of the modified Kalman filter is that it is tractable to identify.

The Kalman filter uses the previous observations $X_1, \dots, X_t$ to compute an estimate of the state z_t as $\hat{z}_t \triangleq \mathbb{E}\left[z_t \mid \mathcal{F}_{t-1}\right]$ where $\mathcal{F}_{t-1}$ is the sigma algebra generated by the rewards $X_1, \dots, X_{t-1}$,

$$\begin{cases} \hat{z}_{t+1} & = \Gamma \hat{z}_t + \Gamma K_t \left(X_t - \langle c_{A_t}, \hat{z}_t \rangle\right), \quad P_{t+1} = g\left(P_t, c_{A_t}\right) \\ K_t & = P_t c_{A_t} \left(c_{A_t}^\top P_t c_{A_t} + \sigma\right)^{-1} \\ \hat{X}_t & = \langle c_{A_t}, \hat{z}_t \rangle \end{cases}, \tag{3}$$

and $g\left(P, c\right)$ is defined to be the following Riccati equation (Gelb et al., 1974)

$$g\left(P, c\right) \triangleq \Gamma P \Gamma^\top + Q - \Gamma P c \left(c^\top P c + \sigma\right)^{-1} c^\top P \Gamma^\top. \tag{4}$$

We impose the following assumption for the LGDS's (1) initial state $z_0 \sim \mathcal{N}(\hat{z}_0, P_0)$ and the Kalman filter's (3) initial error covariance matrix P_0:

Assumption 4 *The initial state $z_0 \in \mathbb{R}^d$ of the LGDS (1) is sampled from a normal distribution with a mean $\hat{z}_0 \in \mathbb{R}^d$ that is a solution of $\hat{z}_0 = \Gamma \hat{z}_0$ and covariance matrix $P_0 \in \mathbb{R}^{d \times d}$. We assume that $P_0 = P_{\overline{a}}$, where $P_{\overline{a}}$ is the steady-state error covariance matrix, $P_{\overline{a}} = g\left(P_{\overline{a}}, c_{\overline{a}}\right)$, $c_{\overline{a}} \in \mathcal{A}$. This assumption implies that the LGDS (1) is in a steady-state distribution.*

Remark 1 *Assumption 4 states that LGDS (1) is in steady-state and the Kalman filter's (3) error covariance matrix is bounded. This is a reasonable assumption as the Kalman filter covariance matrix P_t converges exponentially to the steady state covariance matrix $P_{\overline{a}}$ as t increases if action $c_{\overline{a}} \in \mathcal{A}$ is consistently chosen. In addition, a similar assumption has been made in Deistler et al. (1995), Knudsen (2001), and Tsiamis and Pappas (2019). Finally, it will be proven in Lemma 2 that there exists an action $c_{\overline{a}} \in \mathcal{A}$ such that $P_{\overline{a}} \succeq P_t$ if $P_{\overline{a}} = P_0$.*

As mentioned earlier, the parameters of LGDS (1) are unknown due to Assumption 2. Therefore, we propose to learn the Kalman filter (3) for reward prediction. However, since the Kalman filter matrices P_t and K_t change constantly, it is intractable to identify the Kalman filter. Therefore, we prove that there exists a modified Kalman filter that has a bounded reward prediction error regardless of the choices $c_{A_t} \in \mathcal{A}$ that is tractable to identify. For proving Theorem 3, we first provide Lemma 2 for the bound on the Kalman filter error covariance matrix P_t.

Lemma 2 *Let P_a, $a \in [k]$ be the steady state solution of the Kalman filter for each action $c_a \in \mathcal{A}$, $P_a = g\left(P_a, c_a\right)$, where $g\left(P_a, c_a\right)$ is defined in (4). Define $P_{\overline{a}} \succeq 0$ to be the steady-state error covariance matrix of the Kalman filter (3) associated with action $c_{\overline{a}} \in \mathcal{A}$ such that $P_{\overline{a}} \succeq P_a$ for every action $a \in [k]$. By imposing Assumptions 1, 3, and 4, the LGDS (1), then $P_{\overline{a}} \succeq P_t$ for any $t = 1, 2, \dots, n$.*

Below is Theorem 3 which proves the existence of a modified Kalman filter with a bounded prediction error. Proof for Theorem 3 can be found in Appendix B of Gornet and Sinopoli (2024).

Theorem 3 *We define the following modified Kalman filter*

$$\begin{cases} \hat{z}'_{t+1} & = \Gamma \hat{z}'_t + \Gamma L_{A_t} \left(X_t - \langle c_{A_t}, \hat{z}'_t \rangle\right) \\ X_t & = \langle c_{A_t}, \hat{z}'_t \rangle + \gamma_{A_t} \end{cases}, \quad L_{A_t} \triangleq P_{\overline{a}} c_{A_t} \left(c_{A_t}^\top P_{\overline{a}} c_{A_t} + \sigma\right)^{-1}, \tag{5}$$

where $\gamma_{A_t} \triangleq X_t - \langle c_{A_t}, \hat{z}'_t \rangle \sim \mathcal{N}\left(0, c_{A_t}^\top P'_t c_{A_t} + \sigma\right)$ *and* $P'_t \triangleq \mathbb{E}\left[\left(z_t - \hat{z}'_t\right)\left(z_t - \hat{z}'_t\right)^\top \mid \mathcal{F}_{t-1}\right]$. *It is proven for the modified Kalman filter* (5) *that 1) the matrix* $\Gamma - \Gamma L_{A_t} c_{A_t}^\top$ *is stable and 2) the variance of the residual Var* (γ_{A_t}) *is bounded.*

The key takeaway for Theorem 3 is that there exists a modified Kalman filter (5) that is easier to identify in comparison to the Kalman filter (3) at the expense of a higher prediction error Var $(\gamma_a) \geq c_a^\top P_t c_a + \sigma^2$. This is because the modified Kalman filter has only a finite number of gain matrices L_{A_t} and a static covariance matrix $P_{\overline{a}}$. In addition, the variance of the prediction error Var (γ_a) has an upper-bound.

3.1. Learning the Modified Kalman filter

Using Theorem 3 and inspired by the results presented in Tsiamis and Pappas (2019), we will learn the modified Kalman filter since the matrices and vectors in the LGDS (1) and its modified Kalman filter (5) are unknown. Let parameter $s > 0$ denote how far in the past the learner will look. We define the tuple $\mathbf{c} \triangleq \begin{pmatrix} c_{A_{t-s}} & \dots & c_{A_{t-1}} \end{pmatrix}$ as the sequence of actions chosen by the learner from rounds $t-s$ to $t-1$. The reward $X_t = \langle c_a, z_t \rangle + \eta_t$ for action $a \in [k]$ can be expressed as a linear combination of rewards $X_{t-s}, \dots, X_{t-1}$ generated by the tuple $\mathbf{c}$ using the matrices defined in the modified Kalman filter (5):

$$\begin{aligned} X_t = c_a^\top \left(\Gamma - \Gamma L_{A_{t-1}}\right) \cdots \left(\Gamma - \Gamma L_{A_{t-s+1}}\right) \Gamma L_{A_{t-s}} X_{t-s} + \dots \\ + c_a^\top \Gamma L_{A_{t-1}} X_{t-1} + c_a^\top \left(\Gamma - \Gamma L_{A_{t-1}} c_{A_{t-1}}^\top\right) \cdots \left(\Gamma - \Gamma L_{A_{t-s}} c_{A_{t-s}}^\top\right) \hat{z}'_{t-s} + \gamma_a, \end{aligned}$$

Therefore, let there be defined the vectors $G_{c_a|\mathbf{c}}$, $c_a \in \mathcal{A}$, and $\Xi\left(\mathbf{c}\right)$ to express the reward $X_t = \langle c_a, z_t \rangle + \eta_t$:

$$\Rightarrow X_t = G_{c_a|\mathbf{c}}^\top \Xi_t\left(\mathbf{c}\right) + \beta_a + \gamma_a, \tag{6}$$

$$\begin{aligned} G_{c_a|\mathbf{c}} &\triangleq \begin{bmatrix} c_a^\top \left(\Gamma - \Gamma L_{A_{t-1}}\right) \cdots \left(\Gamma - \Gamma L_{A_{t-s+1}}\right) \Gamma L_{A_{t-s}} & \dots & c_a^\top \Gamma L_{A_{t-1}} \end{bmatrix} \in \mathbb{R}^{s \times 1} \\ \Xi_t\left(\mathbf{c}\right) &\triangleq \begin{bmatrix} X_{t-s} & \dots & X_{t-1} \end{bmatrix}^\top \in \mathbb{R}^{s \times 1} \\ \beta_a &\triangleq c_a^\top \left(\Gamma - \Gamma L_{A_{t-1}} c_{A_{t-1}}^\top\right) \cdots \left(\Gamma - \Gamma L_{A_{t-s}} c_{A_{t-s}}^\top\right) \hat{z}'_{t-s} \in \mathbb{R}. \end{aligned}$$

Based on equation (6), we can express the reward $X_t = \langle c_a, z_t \rangle + \eta_t$ for each action $c_a \in \mathcal{A}$ using $G_{c_a|\mathbf{c}}$, $c_a \in \mathcal{A}$, and $\Xi\left(\mathbf{c}\right)$ with the following linear model:

$$\begin{pmatrix} \langle c_1, z_t \rangle + \eta_t \\ \langle c_2, z_t \rangle + \eta_t \\ \vdots \\ \langle c_k, z_t \rangle + \eta_t \end{pmatrix} = \begin{pmatrix} G_{c_1|\mathbf{c}}^\top \\ G_{c_2|\mathbf{c}}^\top \\ \vdots \\ G_{c_k|\mathbf{c}}^\top \end{pmatrix} \Xi_t\left(\mathbf{c}\right) + \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_k \end{pmatrix}. \tag{7}$$

The linear model (7) proves that we only need to identify k^{s+1} vectors $G_{c_a|\mathbf{c}}$. Therefore, we can 1) identify $G_{c_a|\mathbf{c}}$ for each action $c_a \in \mathcal{A}$ and 2) predict the reward X_t using inner product of the identified $G_{c_a|\mathbf{c}}$ and sequence of rewards $X_{t-s}, \dots, X_{t-1}$.

Remark 4 *In the linear model (7), there is a parameter $s > 0$ which is the number of previous rewards $X_{t-s}, \ldots, X_{t-1}$ used for predicting the reward X_t. Parameter $s > 0$ impacts the magnitude of the term β_a (which decreases exponentially as s increases) and the number of linear models $G_{c_a|\mathbf{c}}$ to identify (which increases exponentially as s increases).*

The following are assumed about $G_{c_a|\mathbf{c}}$, γ_a, and $c_a \in \mathcal{A}$:

Assumption 5 *There exists a known upper bound B_G such that $\|G_{c_a|\mathbf{c}}\|_2 \leq B_G$ for all $a \in [k]$ which is a common assumption to use in SMAB problems (Lattimore and Szepesvári, 2020).*

Assumption 6 *There exists a known constant $B_R > 0$ such that for any round $t > 0$, we have:*

$$\sqrt{tr\left(Z_t\right)} \leq B_R,\ Z_t \triangleq \mathbb{E}\left[z_t z_t^\top\right],\ Var\left(\gamma_a\right) \leq c_a^\top P_{\overline{a}} c_a + \sigma_\eta \leq B_R^2 \text{ for } c_a \in \mathcal{A},$$

where Z_t (which has the iteration $Z_{t+1} = \Gamma Z_t \Gamma^\top + Q$) is the covariance of the LGDS's (1) state z_t. Results in the area of non-stationary SMAB have made similar assumptions (see Chen et al. (2023)).

To learn $G_{c_a|\mathbf{c}}$, assume that at time points $\mathcal{T}_{c_a|\mathbf{c}} = \{t_1, \ldots, t_{N_a}\}$ (N_a is the number of times action $c_a \in \mathcal{A}$ is chosen) the following tuple sequence $\left(c_{A_{t_i-s}}, \ldots, c_{A_{t_i-1}}\right) = \mathbf{c} \in \mathcal{A}^s$ for $t_i \in \{t_1, \ldots, t_{N_a}\}$ and action $c_{A_t} = c_a \in \mathcal{A}$ are chosen. We have the following linear model

$$\mathbf{X}_{\mathcal{T}_{c_a|\mathbf{c}}} = G_{c_a|\mathbf{c}}^\top \mathbf{Z}_{\mathcal{T}_{c_a|\mathbf{c}}} + \mathbf{B}_{\mathcal{T}_{c_a|\mathbf{c}}} + \mathbf{E}_{\mathcal{T}_{c_a|\mathbf{c}}}, \tag{8}$$

$$\mathbf{X}_{\mathcal{T}_{c_a|\mathbf{c}}} \triangleq \begin{bmatrix} X_{t_1} & \ldots & X_{t_{N_a}} \end{bmatrix} \in \mathbb{R}^{1\times N_a}, \qquad \mathbf{Z}_{\mathcal{T}_{c_a|\mathbf{c}}} \triangleq \begin{bmatrix} \Xi_{t_1}(\mathbf{c}) & \ldots & \Xi_{t_{N_a}}(\mathbf{c}) \end{bmatrix} \in \mathbb{R}^{s\times N_a},$$
$$\mathbf{B}_{\mathcal{T}_{c_a|\mathbf{c}}} \triangleq \begin{bmatrix} \beta_{A_{t_1}} & \ldots & \beta_{A_{t_{N_a}}} \end{bmatrix} \in \mathbb{R}^{1\times N_a}, \qquad \mathbf{E}_{\mathcal{T}_{c_a|\mathbf{c}}} \triangleq \begin{bmatrix} \gamma_{A_{t_1}} & \ldots & \gamma_{A_{t_{N_a}}} \end{bmatrix} \in \mathbb{R}^{1\times N_a}.$$

The least squares estimate of $G_{c_a|\mathbf{c}}$ in (8) is

$$\hat{G}_{c_a|\mathbf{c}}\left(\mathcal{T}_{c_a|\mathbf{c}}\right) = \mathbf{X}_{\mathcal{T}_{c_a|\mathbf{c}}} \mathbf{Z}_{\mathcal{T}_{c_a|\mathbf{c}}}^\top V_a\left(\mathcal{T}_{c_a|\mathbf{c}}\right)^{-1} \tag{9}$$

$$V_a\left(\mathcal{T}_{c_a|\mathbf{c}}\right) \triangleq \lambda I_s + \mathbf{Z}_{\mathcal{T}_{c_a|\mathbf{c}}} \mathbf{Z}_{\mathcal{T}_{c_a|\mathbf{c}}}^\top = \lambda I_s + \sum_{i=1}^{N_a} \Xi_{t_i}(\mathbf{c}) \Xi_{t_i}(\mathbf{c})^\top, \tag{10}$$

where $\lambda > 0$ is a regularization term. Since there are k^s codes $\mathbf{c} \in \mathcal{A}^s$, then there are k^{s+1} vectors $G_{c_a|\mathbf{c}}$ to learn.

4. Uncertainty-Based System Search Restless Bandit Problem

The section above provided a predictor, the modified Kalman filter, for the rewards generated by the LGDS (1). It also provided a methodology for identifying the predictor. Now that the reward can be predicted using an identified modified Kalman filter, we discuss how to use the predictor in Algorithm 1, Uncertainty-Based System Search (UBSS). The general scheme for UBSS is to 1) identify the predictor $G_{c_a|\mathbf{c}}$ for each action $c_a \in \mathcal{A}$ and 2) select actions that balances what the learner predicts will return the highest reward versus which actions the learner is the most uncertain

due to the error of the predictor $\hat{G}_{c_a|\mathbf{c}}$. Therefore, for each round t in UBSS, the learner will choose actions based on the following optimization problem

$$\arg\max_{a\in[k]} \hat{G}_{c_a|\mathbf{c}} \left(\mathcal{T}_{c_a|\mathbf{c}}\right)^\top \Xi_t(\mathbf{c}) + \left(e_{c_a|\mathbf{c}}(\delta_e) + b_{c_a|\mathbf{c}}(\delta_b)\right) \sqrt{\Xi_t(\mathbf{c})^\top V_a \left(\mathcal{T}_{c_a|\mathbf{c}}\right)^{-1} \Xi_t(\mathbf{c})}, \quad (11)$$

where with a probability of at least $(1-\delta_e)(1-\delta_b)$, the following inequality is satisfied (see Theorem 9 in Appendix C.2 of Gornet and Sinopoli (2024)):

$$\hat{G}_{c_a|\mathbf{c}} \left(\mathcal{T}_{c_a|\mathbf{c}}\right)^\top \Xi_t(\mathbf{c}) - G_{c_a|\mathbf{c}}^\top \Xi_t(\mathbf{c}) \leq \left(e_{c_a|\mathbf{c}}(\delta_e) + b_{c_a|\mathbf{c}}(\delta_b)\right) \sqrt{\Xi_t(\mathbf{c})^\top V_a \left(\mathcal{T}_{c_a|\mathbf{c}}\right)^{-1} \Xi_t(\mathbf{c})}. \quad (12)$$

The terms $e_{c_a|\mathbf{c}}(\delta_e)$ and $b_{c_a|\mathbf{c}}(\delta_b)$ are defined as

$$e_{c_a|\mathbf{c}}(\delta_e) \triangleq \sqrt{2B_R^2 \log\left(\frac{1}{\delta_e} \frac{\det(V_a\left(\mathcal{T}_{c_a|\mathbf{c}}\right))^{1/2}}{\det(\lambda I)^{1/2}}\right)} \quad (13)$$

$$b_{c_a|\mathbf{c}}(\delta_b) \triangleq \sqrt{N_a} \frac{B_c B_R}{\delta_b} \sqrt{\mathrm{tr}\left(I - \lambda V_a \left(\mathcal{T}_{c_a|\mathbf{c}}\right)^{-1}\right)} + \lambda \sqrt{\mathrm{tr}\left(V_a \left(\mathcal{T}_{c_a|\mathbf{c}}\right)^{-1}\right)} B_G. \quad (14)$$

Reward prediction uncertainty (12) of action $c_a \in \mathcal{A}$ is impacted directly $V_a\left(\mathcal{T}_{c_a|\mathbf{c}}\right)$ which is a sum of N_a (number of times action $c_a \in \mathcal{A}$ is chosen) positive semi-definite matrices. Therefore, choosing an action frequently (large N_a) will lower the reward prediction uncertainty. The rationale behind optimization problem (11) is to balance choosing the action with the highest reward versus the action with the most uncertainty. We summarize below which term is defined to be in (11) within Algorithm 1 which consists of an **Exploitation term** (which action the learner expects to return the highest reward) and an **Exploration term** (how much should the learner explore an action).

- **Exploitation term**: $\hat{G}_{c_a|\mathbf{c}} \left(\mathcal{T}_{c_a|\mathbf{c}}\right)^\top \Xi_t(\mathbf{c})$
- **Exploration term**: $\left(e_{c_a|\mathbf{c}}(\delta_e) + b_{c_a|\mathbf{c}}(\delta_b)\right) \sqrt{\Xi_t(\mathbf{c})^\top V_a \left(\mathcal{T}_{c_a|\mathbf{c}}\right)^{-1} \Xi_t(\mathbf{c})}$

Parameters (δ_e, δ_b) are failure rates of the bound in (12) where (δ_e, δ_b) values closer to 0 computes a larger bound (12). Parameter s is the number of previously observed rewards $X_{t-\tau}$ $(\tau = 1, \ldots, s)$ used for predicting the next reward. The number of models to learn increases exponentially as s increases. Finally, λ is a regularization parameter to ensure that (10) is invertible.

4.1. Regret Performance

Algorithm 1, UBSS, has the following upper bound on regret (2). Proof is in Appendix C.3 of Gornet and Sinopoli (2024).

Theorem 5 *Using Algorithm 1 and setting $\delta_e = \delta_b = \delta \in (0,1)$, regret (2) satisfies the following inequality with a probability of at least $(1-\delta)^4$:*

$$R_n \leq \sum_{t=1}^{s} \max_{c_a \in \mathcal{A}} \mathbb{E}\left[\langle c_{a^*} - c_a, z_t \rangle\right] + \sum_{a=1}^{k} 2(n-s) B_c^2 B_R^2 \left(1 - (1-\delta)^4 \left(1 - \exp\left(\frac{-4B(\delta \mid \mathbf{c})^2}{2\Delta G_{c_a|\mathbf{c}}^\top \Sigma_{\Xi_t(\mathbf{c})} \Delta G_{c_a|\mathbf{c}}}\right)\right)\right), \quad (15)$$

Algorithm 1: Uncertainty-Based System Search (UBSS)

Input: $\delta_e, \delta_b \in (0,1)$, $\lambda > 0$, $s \in \mathbb{N}$, $B_c, B_R, B_G \in \mathbb{R}^+$

```
// Initialization
```

for $\mathbf{c} \in \left\{ \begin{pmatrix} c_{a_1} & c_{a_2} & \dots & c_{a_s} \end{pmatrix} \in \mathcal{A}^s \right\}$ **do**

  for $a \in [k]$ **do**

    $\mathcal{T}_{c_a|\mathbf{c}} \leftarrow \{\}$, $V_a\left(\mathcal{T}_{c_a|\mathbf{c}}\right) \leftarrow \lambda I_s$, $\hat{G}_{c_a|\mathbf{c}}\left(\mathcal{T}_{c_a|\mathbf{c}}\right) \leftarrow \mathbf{0}_{s\times 1}$

    $\left(e_{c_a|\mathbf{c}}\left(\delta_e\right), b_{c_a|\mathbf{c}}\left(\delta_b\right)\right) \leftarrow 1/\epsilon$ where ϵ small

  end

end

```
// Learner interaction with LGDS
```

for $t = 1, 2, \dots, n$ **do**

  if $t \geq s$ **then**

    ```
    // Action selection
    ```

    $A_t \leftarrow \underset{a\in\{1,2,\dots,k\}}{\arg\max}\ \hat{G}_{c_a|\mathbf{c}}\left(\mathcal{T}_{c_a|\mathbf{c}}\right)^\top \Xi_t(\mathbf{c}) + \left(e_{c_a|\mathbf{c}}\left(\delta_e\right) + b_{c_a|\mathbf{c}}\left(\delta_b\right)\right)\sqrt{\Xi_t(\mathbf{c})^\top V_a\left(\mathcal{T}_{c_a|\mathbf{c}}\right)^{-1}\Xi_t(\mathbf{c})}$

    Sample X_t from (1)

    $\mathbf{c} \leftarrow \begin{pmatrix} c_{A_{t-s}} & \dots & c_{A_{t-1}} \end{pmatrix}$

    $\mathcal{T}_{c_{A_t}|\mathbf{c}} \leftarrow \mathcal{T}_{c_{A_t}|\mathbf{c}} \cup \{t\}$

    ```
    // Update estimates
    ```

    $V_{A_t}\left(\mathcal{T}_{c_{A_t}|\mathbf{c}}\right) \leftarrow V_{A_t}\left(\mathcal{T}_{c_{A_t}|\mathbf{c}}\right) + \Xi_t(\mathbf{c})\,\Xi_t(\mathbf{c})^\top$

    $\hat{G}_{c_{A_t}|\mathbf{c}}\left(\mathcal{T}_{c_{A_t}|\mathbf{c}}\right) \leftarrow \hat{G}_{c_{A_t}|\mathbf{c}}\left(\mathcal{T}_{c_{A_t}|\mathbf{c}}\right)^\top + X_t \Xi_t(\mathbf{c})^\top V_{A_t}\left(\mathcal{T}_{c_{A_t}|\mathbf{c}}\right)^{-1}$

    ```
    // Update bounds
    ```

    Set $e_{c_{A_t}|\mathbf{c}}(\delta)$ and $b_{c_{A_t}|\mathbf{c}}(\delta)$ based on (13) and (14), respectively.

  else

    $A_t \leftarrow$ Sample uniformly $a \sim [k]$

    Sample X_t from (1)

  end

end

where $\Delta G_{c_a|\mathbf{c}} \triangleq G_{c_{a^*}|\mathbf{c}} - G_{c_a|\mathbf{c}}$, $\Sigma_{\Xi_t(\mathbf{c})} \triangleq \mathbb{E}\left[\Xi_t(\mathbf{c})\,\Xi_t(\mathbf{c})^\top\right]$, *and* $B(\delta \mid \mathbf{c})$ *is*

$$B(\delta \mid \mathbf{c}) \triangleq \sqrt{2B_R^2 \log\left(\frac{1}{\delta}\frac{\left(s\lambda + (n-s)\frac{\mathbb{E}\left[\|\Xi_t(\mathbf{c})\|_2\right]}{\delta}\right)^{s/2}}{\lambda^{s/2}}\right)}\sqrt{\frac{s}{\lambda}}\frac{\mathbb{E}\left[\|\Xi_t(\mathbf{c})\|_2\right]}{\delta} + \sqrt{n-s}\frac{B_c B_R}{\delta}\sqrt{s}\sqrt{\frac{s}{\lambda}}\frac{\mathbb{E}\left[\|\Xi_t(\mathbf{c})\|_2\right]}{\delta} + \lambda B_G\sqrt{\frac{s}{\lambda}}\frac{\mathbb{E}\left[\|\Xi_t(\mathbf{c})\|_2\right]}{\delta}. \quad (16)$$

Theorem 5 proves that regret increases at worst linearly, i.e. $\mathcal{O}(n)$. If the covariance for two different actions is large, e.g. large $\Delta G_{c_a|\mathbf{c}}$ and $\Sigma_{\Xi_t(\mathbf{c})}$ terms, then the bound will decrease. The bound decreases exponentially as uncertainty (12) decreases.

5. Numerical Results

For numerical results, we generated rewards $X_t \in \mathbb{R}$ for each action $\{c_1, c_2\}$ from the following LGDS:

$$\begin{cases} z_{t+1} &= \begin{pmatrix} 0.9R(\theta) & I_2 \\ \mathbf{0}_{2\times 2} & 0.9R(\theta) \end{pmatrix} z_t + \xi_t \\ X_t &= \langle c_{A_t}, z_t \rangle + \eta_t \end{pmatrix}, \quad \begin{cases} R(\theta) &\triangleq \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \\ c_1 &\triangleq \begin{pmatrix} 10 & 0 & 0 & 0 \end{pmatrix} \\ c_2 &\triangleq \begin{pmatrix} 0 & 10 & 0 & 0 \end{pmatrix} \end{cases}, \tag{17}$$

where the process noise $\xi_t \sim \mathcal{N}(0, I_4)$ and the measurement noise $\eta_t \sim \mathcal{N}(0, 1)$ are sampled from standard normal distributions. The LGDS (17) is proposed to study how the magnitude of the error covariance matrix $P_{\bar{a}}$ impacts performance of UBSS, where $P_{\bar{a}}$ is directly impacted by the parameter $\theta \in [0, 2\pi]$. Prior to the learner's interaction with the LGDS (17), 10^4 time steps are computed of the LGDS (17) to set the system to a steady state. After, the 10^4 time steps, the length of the interaction between the environment and the learner is $n = 10^4$ rounds. Regret (2) is used to provide a metric of performance. Parameter s in Algorithm 1, UBSS, is set to 1 in the top left plot. For comparison, we consider UCB (Agrawal, 1995), SW-UCB (Garivier and Moulines, 2008), and a learner that selects a random action each round (this learner is denoted as Random). We use UCB as a comparison since the eigenvalues of the LGDS (17) state matrix $\Gamma \in \mathbb{R}^{4\times 4}$ is Schur, implying that the reward distributions have a bounded covariance with a mean of zero. SW-UCB is also used as a comparison since the reward is still generated by a dynamical system. Finally, Random is used as baseline for worst performance.

In the top left plot of Figure 1, the percentage of UBSS's regret (2) is lower than UCB (red), Sliding UCB (green) and Random (blue) regret is shown for each $\theta \in [0, 2\pi]$. The middle plot of Figure 1 is the minimum eigenvalue of the Observability Gramian $\mathcal{O}$ (Hespanha, 2018) for both actions $c_a \in \{c_1, c_2\}$, which is the solution of the Lyapunov equation $\mathcal{O} = \Gamma^\top \mathcal{O} \Gamma + c_a c_a^\top$. The bottom plot of Figure 1 is the real part of the eigenvalue of the state matrix Γ. In the white regions, all the comparison algorithms outperform UBSS. Based on the plot in the middle, it appears that the low Observability Gramian minimum eigenvalue and a positive real part of the state matrix's eigenvalue is the cause. For the blue regions, no algorithm outperforms Random, implying that the rewards are too noisy to estimate/predict for the compared algorithms. Finally, the gray regions is approximately where UBSS performs the best, providing approximately a 10% improvement for each of the algorithms mid-region. Based on the bottom 2 plots, this increase in performance is from the high observability and an eigenvalue with a negative real part for the state matrix. High observability lowers the magnitude of the error covariance matrix $P_{\bar{a}}$, which leads to a lower regret bound of UBSS. In addition, an eigenvalue with a negative real part for the state matrix leads to rapid switching of the optimal action, making it difficult for UCB to adapt. For the plot on the far right, this is the relative performance of UBSS for each parameter $s = 1, 2, 3$ when the LGDS system (17) parameter set to approximately $5\pi/8$ (approximately where we see the largest improvement in performance of UBSS in the top left plot). Therefore, it appears that as s increases to $s = 2, 3$, regret

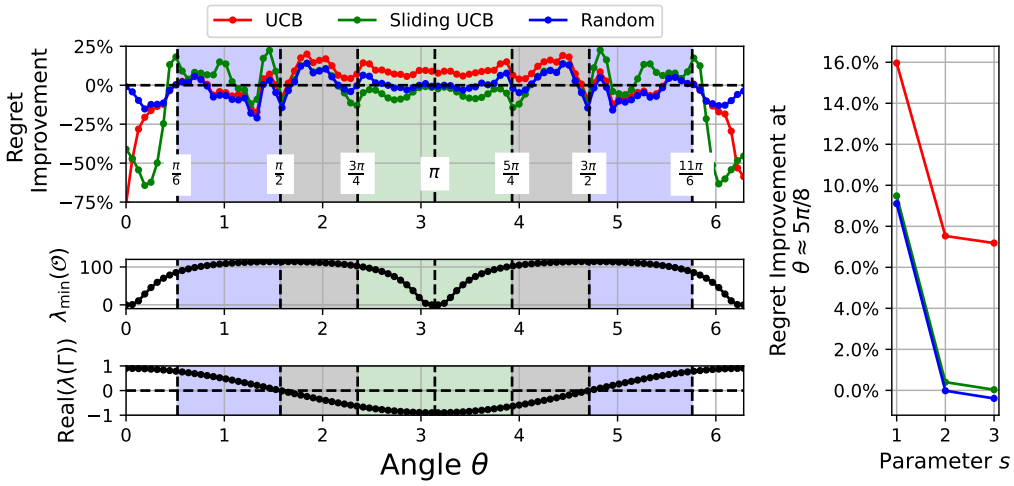

Figure 1: Comparison algorithm's regret normalized with respect to UBSS's regret. A positive percent implies that UBSS has a lower regret than the compared algorithm.

performance of UBSS decreases. Since the number of parameters to identify increases exponentially as s increases (leading to longer exploration times), regret performance of UBSS decreases as s increases.

6. Conclusion

We have presented an algorithm for addressing a variation of the restless bandit with a continuous state-space. The rewards generated by this restless bandit variation is a LGDS. Based on the formulation, we propose to learn a representation of the modified Kalman filter to predict the rewards for each action. We have shown that regardless of the sequence of actions chosen, the learned representation of the modified Kalman filter converges. It is then proven what strategy should be used given the bound on regret, leading to an uncertainty-based strategy.

In this work, we have not considered how the sequence of actions impact prediction error, how to choose window size (how far the learner looks into the past), and best obtainable performance of SMAB with LGDS environments. First, the perturbation added for exploration only considers error of the model and not the sequence of actions impact on the error of the prediction. In other words, the chosen sequence of actions are myopic. Therefore, future work will focus on the action sequence impact on the reward prediction. Next, an important parameter in UBSS is the window size. In UBSS, this is a parameter to set prior to the interaction with the environment. However, questions we care to ask is how to automate the process of choosing window size. Finally, UBSS has linear regret performance. Therefore, future work will be to derive the best obtainable performance of *any* algorithm applied to a SMAB with rewards generated by this paper's proposed LGDS. We will then analyze if UBSS regret performance is close or far to the best obtainable performance.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Rajeev Agrawal. Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems*, 27:199–207, 2014.

Ilija Bogunovic, Jonathan Scarlett, and Volkan Cevher. Time-varying gaussian process bandit optimization. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 314–323, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/bogunovic16.html.

Qinyi Chen, Negin Golrezaei, and Djallel Bouneffouf. Non-stationary bandits with auto-regressive temporal dependency. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Wenhan Dai, Yi Gai, Bhaskar Krishnamachari, and Qing Zhao. The non-bayesian restless multi-armed bandit: A case of near-logarithmic regret. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2940–2943. IEEE, 2011.

Manfred Deistler, K Peternell, and Wolfgang Scherrer. Consistency and relative efficiency of subspace methods. *Automatica*, 31(12):1865–1875, 1995.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008.

Arthur Gelb et al. *Applied optimal estimation*. MIT press, 1974.

Jonathan Gornet and Bruno Sinopoli. Restless bandit problem with rewards generated by a linear gaussian dynamical system. *arXiv preprint arXiv:2405.09584*, 2024.

Jonathan Gornet, Mehdi Hosseinzadeh, and Bruno Sinopoli. Stochastic multi-armed bandits with non-stationary rewards generated by a linear dynamical system. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1460–1465. IEEE, 2022.

Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018.

Young Hun Jung and Ambuj Tewari. Regret bounds for thompson sampling in episodic restless bandit problems. *Advances in Neural Information Processing Systems*, 32, 2019.

Torben Knudsen. Consistency analysis of subspace identification methods based on a linear regression approach. *Automatica*, 37(1):81–89, 2001.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Haoyang Liu, Keqin Liu, and Qing Zhao. Logarithmic weak regret of non-bayesian restless multi-armed bandit. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1968–1971. IEEE, 2011.

Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret bounds for restless markov bandits. In *International conference on algorithmic learning theory*, pages 214–228. Springer, 2012.

Jack Parker-Holder, Vu Nguyen, and Stephen J Roberts. Provably efficient online hyperparameter optimization with population-based bandits. *Advances in neural information processing systems*, 33:17200–17211, 2020.

Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits. In *COLT*, pages 343–354, 2008.

Cem Tekin and Mingyan Liu. Online learning of rested and restless bandits. *IEEE Transactions on Information Theory*, 58(8):5588–5611, 2012.

Anastasios Tsiamis and George J Pappas. Finite sample analysis of stochastic system identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3648–3654. IEEE, 2019.

Siwei Wang, Longbo Huang, and John Lui. Restless-ucb, an efficient and low-complexity algorithm for online restless bandits. *Advances in Neural Information Processing Systems*, 33: 11878–11889, 2020.

Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A):287–298, 1988.