Switching Latent Bandits

Anonymous authors
Paper under double-blind review

Abstract

We consider a Latent Bandit problem where the latent state keeps changing in time according to an underlying Markov Chain and every state is represented by a specific Bandit instance. At each step, the agent chooses an arm and observes a random reward but is unaware of which MAB he is currently pulling. As typical in Latent Bandits, we assume to know the reward distribution of the arms of all the Bandit instances. Within this setting, our goal is to learn the transition matrix determined by the Markov process, so as to minimize the cumulative regret. We propose a technique to solve this estimation problem that exploits the properties of Markov Chains and results in solving a system of linear equations. We present an offline method that chooses the best subset of possible arms that can be used for matrix estimation, and we ultimately introduce the SL-EC learning algorithm based on an Explore Then Commit strategy that builds a belief representation of the current state and optimizes the instantaneous regret at each step. This algorithm achieves a regret of the order $\widetilde{\mathcal{O}}(T^{2/3})$ with T being the considered horizon. Finally, we illustrate the effectiveness of the approach and compare it with state-of-the-art algorithms for non-stationary bandits.

1 Introduction

The Multi-Armed Bandit (MAB) framework is a well-known model used for sequential decision-making with little or no information. This framework has been successfully applied in a large number of fields, such as recommender systems, advertising, and networking. In the general MAB formulation, a learner sequentially selects an action among a finite set of different ones. The choice over the arm to select is made by properly balancing the exploration-exploitation trade-off with the goal of maximizing the expected total reward over a horizon T and guaranteeing the *no-regret* property, thus meaning that the loss incurred by not knowing the best arm is increasing sublinearly over time. Standard MAB literature requires the payoff of the available actions to be stationary (i.e., rewards come from a fixed distribution) in order to design efficient no-regret algorithms.

However, in many real-life applications, the stationarity assumption may not necessarily hold as data may be subjected to changes over time. In some applications, it is also possible to identify different data distributions each one corresponding to a specific working regime. In cases of large availability of historical data appearing in the form of past user interactions, it is possible to learn *offline* the observation models associated with the different arms for each working regime. Exploiting the knowledge on observation models leads to many advantages over the *fully online exploration* setting where no prior information is available at the beginning and a massive number of interactions is required to learn the observation models associated with each working regime. Even if the latent regime is not directly observable, by knowing the observation distributions, it can be inferred from the interaction process. Identifying the latent state accelerates the adaptation of the agent to the environment leading to improved performances over time.

Past works focused on this state identification problem under the assumption of knowing the conditional observation models (Maillard & Mannor, 2014; Zhou & Brunskill, 2016) and defined theoretically optimal UCB algorithms. Follow-up work of Hong et al. (2020a) provided more practical Thompson Sampling algorithms also considering the problem of model misspecification and came up with an analysis on the Bayes regret.

The works cited above assume that the latent state does not change during the interaction process: once the real state is identified, the agent can act optimally. Differently, in this work, we embrace a more realistic scenario and assume that the latent state can change through time. In accordance with the latent bandits setting, we assume that the learning agent is aware of the observation models of the arms conditioned on each latent state. A setting similar to ours has been considered also in Hong et al. (2020b), the key difference is that they assume to have either full or partial knowledge of both the observation model and the transition model. We instead focus on the more challenging problem of learning the transition model given the knowledge of the observation models and maximizing the cumulative reward over T interaction steps. More specifically, our problem is modeled by assuming the existence of a set $\mathbb{S}$ of different MABs all sharing the same set of finite arms $\mathbb{I}$, each generating rewards (observations) in a finite set $\mathbb{V}$. Each state $s \in \mathbb{S} = \{s_1, \ldots, s_S\}$ represents a different instance of a MAB. At each time step t, there is a transition from latent state s_{t-1} to the new latent state s_t according to the transition matrix governing the process. The action a_t selected in t will thus generate a reward conditioned on the latent state s_t.

Our Contribution We summarize here the main aspects and contributions related to this work:

- we design a procedure for the estimation of the transition matrix that converges to the true value under some mild assumptions. In order to obtain this result, we exploit the information derived from the conditional reward models, and we use some properties of Markov Chains;
- we provide high-probability confidence bounds for the proposed procedure using known results from statistical theory and novel estimation bounds of samples coming from Markov Chains;
- we propose the *Switching Latent Explore then Commit* (SL-EC) algorithm that uses the presented estimation method and then exploits the learned information achieving a $\widetilde{\mathcal{O}}(T^{2/3})$ regret bound on a finite horizon T;
- we illustrate the effectiveness of the approach and compare it with state-of-the-art algorithms for the non-stationary bandits setting.

2 Related Works

Non-stationary Bandits Non-stationary behaviors are closer to real-world scenarios, and this has induced a vast interest in the scientific community leading to the formulation of different methods that consider either abruptly changing environments (Garivier & Moulines, 2011), smoothly changing environments (Trovò et al., 2020), or settings with a bounded variation of the rewards (Besbes et al., 2014). It is known that when rewards may arbitrarily change over time, the problem of Non-Stationary Bandits is intractable, meaning that only trivial bounds can be derived on the dynamic pseudo-regret. That is the main reason why in the literature there is a large focus on non-stationary settings enjoying some specific structure in order to design algorithms with better guarantees. Non-stationary MAB approaches typically include both passive methods in which arm selection is mainly driven by the most recent feedback (Auer et al., 2019; Besbes et al., 2014; Trovò et al., 2020) and active methods where a change detection layer is used to actively perceive a drift in the rewards and to discard old information (Liu et al., 2017; Cao et al., 2018). Works such as Garivier & Moulines (2011) provide a $\mathcal{O}(\sqrt{T})$ regret guarantee under the assumption of knowing the number of abrupt changes. Other works, such as Besbes et al. (2014), employ a fixed budget to bound the total variation of expected rewards over the time horizon. They are able to provide a near-optimal frequentist algorithm with pseudo-regret $\mathcal{O}(T^{2/3})$ and a distribution-independent lower bound. All the above methods are not suited for environments that switch between different regimes as they do not keep in memory past interactions but rather tend to forget or discard the past.
A particular type of non-stationary Bandit problem related to our work includes the *restless Markov* setting (Ortner et al., 2014; Slivkins & Upfal, 2008) where each arm is associated with a different Markov process and the state of each arm evolves independently of the learner's actions. Differently, Fiez et al. (2018) investigate MAB problems with rewards determined by an unobserved Markov Chain where the transition to the next state depends on the action selected at each time step, while Zhou et al. (2021) focus on MAB problems where the state transition dynamics evolves independently of the chosen action. This last work has many similarities with our setting. The main difference lies in the fact that they do not assume to know the conditional reward models and learn them jointly with the transition matrix. They make use

of spectral decomposition techniques (Anandkumar et al., 2014) and use this tool in a regret minimization algorithm achieving a $\mathcal{O}(T^{2/3})$ regret bound. Their setting is more complex than ours but involves stronger assumptions, like the invertibility of the transition matrix that defines the Chain. Furthermore, spectral methods need a vast amount of samples in order to provide reasonable estimation errors and can hardly be used in large problems.

Latent Bandits More similar lines of work are related to bandit studies where latent variables determine the distribution of rewards (Maillard & Mannor, 2014; Zhou & Brunskill, 2016). In these works, the unobserved state is fixed across different rounds and the conditional rewards depend on the latent state. Maillard & Mannor (2014) developed UCB algorithms without context considering the two different cases in which the conditional rewards are either known or need to be estimated. This line of work has been extended to the contextual bandit case in Zhou & Brunskill (2016) where there is an offline procedure to learn the policies and a selection strategy to use them online. Hong et al. (2020a) proposed a TS procedure in the contextual case that updates a prior probability over the set of states in order to give a higher probability to the real latent state. A non-stationary variant of this setting is proposed in Hong et al. (2020b) where the latent states are assumed to change according to a Markov Chain. They develop TS algorithms under different cases when both the reward and transition models are completely known and when partial information about them is available. For the partial information case, they provide an algorithm based on particle filter which will be used for comparison in the experimental section. Differently from Hong et al. (2020b), we do not assume any prior information about the transition matrix and we learn it through interactions with the environment using the information about the reward models.

3 Switching Latent Bandits

3.1 Preliminaries

Markov Chains A Markov Chain (or Markov Process) (Feller, 1968) over the state space $\mathbb{S}$ is a stochastic process $(S_t)_{t=1}^{\infty}$ satisfying the Markov property, meaning that for all $s_i, s_j \in \mathbb{S}$ and $t > 0$:

$$P(S_{t+1} = s_j | S_t = s_i, \dots, S_0 = s_0) = P(S_{t+1} = s_j | S_t = s_i).$$

More formally, a Markov chain is identified by a tuple $\langle \mathbb{S}, \boldsymbol{P}, \boldsymbol{\nu} \rangle$ with $\mathbb{S} = \{s_1, \dots, s_S\}$ being a (finite) set of states, $\boldsymbol{P}$ is a state transition probability matrix with element $P_{ss'} = P(S_{t+1} = s' | S_t = s)$ and $\boldsymbol{\nu} \in \Delta^{\mathbb{S}-1}$ is the initial state distribution with $\nu_s = P(S_0 = s)$. Given the starting distribution $\boldsymbol{\nu}$ and the transition matrix $\boldsymbol{P}$, we can define the probability distribution over the state space after n steps as:

$$\boldsymbol{\nu}^{(n)} = \boldsymbol{\nu} \boldsymbol{P}^n.$$

We can classify Markov Chains according to the different properties they satisfy. In particular, a Markov Chain is *Regular* if some power n of the transition matrix $\boldsymbol{P}^n$ has only positive elements (Puterman, 1994). If a Markov Chain is Regular, it admits a unique stationary distribution, as can be seen in the following:

Proposition 3.1. *Let $\boldsymbol{P}$ be the transition matrix of a Regular Markov Chain and $\boldsymbol{v}$ an arbitrary probability vector. Then:*

$$\lim_{n \to \infty} \boldsymbol{v} \boldsymbol{P}^n = \boldsymbol{\pi},$$

where $\boldsymbol{\pi}$ is the unique stationary distribution of the chain, and the components of the vector $\boldsymbol{\pi}$ are all strictly positive.

Having established the concept of stationary distribution, we give now another core definition, the one of *spectral gap*, that will be useful for what will follow. Before that, we define the set $(\lambda_i)_{i \in [S]}$ of ordered eigenvalues of $\boldsymbol{P}$, with $1 \geq |\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_S|$. Assuming to consider a Regular Markov Chain, the system has a unique stationary distribution, and an eigenvalue $\lambda_1 = 1$.

Definition 3.1. *The spectral gap β of a Markov Process defined by transition matrix $\boldsymbol{P}$ is* $1 - |\lambda_2|$.

The spectral gap provides valuable information about the process. For Regular Markov Chains, the spectral gap controls the rate of exponential decay to the stationary distribution (Saloff-Coste, 1997).

3.2 Problem Formulation

Consider a set $\mathbb{S}$ of $S = |\mathbb{S}|$ different MAB problems. Each MAB has a finite set of discrete arms $\mathbb{I} := \{a_1, \ldots, a_I\}$ with cardinality $I = |\mathbb{I}|$ and, by pulling an arm a, it is possible to get a reward r taken from the set $\mathbb{V} = \{r_1, \ldots, r_V\}$ of possible rewards. In our setting, we assume to have a finite set of rewards $V = |\mathbb{V}|$ with each reward $r \in \mathbb{V}$ bounded for simplicity in the range $[0, 1]$. All the considered MABs share the same sets of arms $\mathbb{I}$ and rewards $\mathbb{V}$. At each step, the MABs alternate according to an underlying Markov Chain having transition probability $\boldsymbol{P}$ with size $S \times S$.
The interaction process is as follows: at each time instant t, the agent chooses an arm $I_t = a$ and observes a reward $R_t = r$ that is determined by the underlying state $S_t = s$ of the process. More formally, the distribution associated with the revealed observation is

$$Q(r|s, a) := P(R_t = r|S_t = s, I_t = a). \tag{1}$$

For the moment, we will stick with the assumption that the distribution $Q(\cdot|s, i)$ is categorical. In Section 5.1, we will see how continuous distributions can also be handled in this setting. Given all the MABs, the actions and possible observations, we can define the three-dimensional observation tensor $\mathbf{O}$ with size $S \times I \times V$ where the element $O_{s,a,r}$ represents the probability of observing the reward r being in state s and pulling arm a.
In particular, by fixing a state s and an action a, the vector $\mathbf{O}_{s,a,:}$ contains the parameters of the categorical distribution associated with state s and action a. Motivated by the realistic scenario of massive availability of past interaction data in domains such as recommender systems that allows learning the reward models during an offline phase, we make the assumption of knowing the observation tensor $\mathbf{O}$ while our objective is to learn the transition matrix $\boldsymbol{P}$ that governs the Chain.

3.3 Reference Matrix Definition

We will introduce here some elements whose utility will be clarified in Section 4.
Let's consider the set $\mathbb{C}_{\mathbb{S}} := \{(s_i, s_j)|s_i, s_j \in \mathbb{S}\}$ with $|\mathbb{C}_{\mathbb{S}}| = S^2$ of all the ordered combinations of pairs of states. These combinations identify all the possible state transitions that can be seen from a generic time step t to the successive one $t+1$. Analogously, we can define the sets $\mathbb{C}_{\mathbb{I}} := \{(a_i, a_j)|a_i, a_j \in \mathbb{I}\}$ with $|\mathbb{C}_{\mathbb{I}}| = I^2$ and $\mathbb{C}_{\mathbb{V}} := \{(r_i, r_j)|r_i, r_j \in \mathbb{V}\}$ with $|\mathbb{C}_{\mathbb{V}}| = V^2$ which are respectively the ordered combinations of pairs of all consecutive arms and of consecutive rewards that can be seen in two contiguous time intervals. From the knowledge of the observation tensor $\mathbf{O}$ and for each $(s_i, s_j) \in \mathbb{C}_{\mathbb{S}}, (a_i, a_j) \in \mathbb{C}_{\mathbb{I}}, (r_i, r_j) \in \mathbb{C}_{\mathbb{V}}$, we are able to compute the following probabilities:

$$P(R_t = r_i, R_{t+1} = r_j|S_t = s_i, S_{t+1} = s_j, I_t = a_i, I_{t+1} = a_j) = O_{s_i,a_i,r_i} O_{s_j,a_j,r_j}. \tag{2}$$

Equation 2 basically allows us to define the probability associated to each possible couple of rewards, actions and states that can occur in consecutive time steps. Hence, by fixing a specific combination of arms (a_h, a_k) from $\mathbb{C}_{\mathbb{I}}$ and by leveraging Equation 2, we can build matrix $\boldsymbol{H}^{a_h,a_k} \in \mathbb{R}^{V^2 \times S^2}$ where the elements along the rows are associated to combinations in $\mathbb{C}_{\mathbb{V}}$ and the elements along the columns are associated to combinations in $\mathbb{C}_{\mathbb{S}}$. The element $H^{a_h,a_k}_{d,e}$ contains the value computed in Equation 2 associated to the d-th combination of rewards in $\mathbb{C}_{\mathbb{V}}$ and the e-th combination of states in $\mathbb{C}_{\mathbb{S}}$ assuming to have pulled actions (a_h, a_k). Having established this procedure to build matrix $\boldsymbol{H}^{a_h,a_k}$ for the couple of actions (a_h, a_k), we can now build similar matrices associated with each of the other combinations of arms. By stacking all these matrices together, we get the matrix $\boldsymbol{A} \in \mathbb{R}^{I^2 V^2 \times S^2}$.
This matrix is a reformulation of the observation tensor $\mathbf{O}$ that expresses the relation between pairs of different elements. The definition of matrix $\boldsymbol{A}$ will be relevant for the proposed estimation method. In the following, we will refer to the matrix $\boldsymbol{A}$ also with the name reference matrix.

3.4 Belief Update

As previously said, at each time step t, we only observe the reward realization, but we are unaware of the Bandit instance from which the arm has been pulled. However, it is possible to define a belief representation

over the current state by using the information derived from the observation tensor $\mathbf{O}$ and the transition matrix $\boldsymbol{P}$ defining the Chain.
We need to introduce a belief vector $\mathbf{b}_t \in \Delta^{S-1}$ representing the probability distribution over the current state at time t. The belief update formulation will follow the typical correction and update step, where the correction step adjusts the current belief $\mathbf{b}_t$ using the reward r_t obtained by pulling arm a_t and the prediction step computes the new belief $\mathbf{b}_{t+1}$ simulating a transition step. The overall update is as follows:

$$\mathrm{b}_{s,t+1} = \frac{\sum_{s'} \mathrm{b}_{s',t} Q(R_t = r_t | S_t = s', I_t = a_t) \boldsymbol{P}(s|s')}{\sum_{s''} Q(R_t = r_t | S_t = s'', I_t = a_t) \mathrm{b}_{s'',t}}. \tag{3}$$

The choice of the arm to pull is driven, at each step t, by

$$I_t = \arg\max_{a \in \mathbb{I}} \sum_{s \in \mathbb{S}} \sum_{r \in \mathbb{V}} r Q(r|s,a) \mathrm{b}_{s,t}. \tag{4}$$

In this case, the goal is to pull the arm that provides the highest instantaneous expected reward, given the belief representation $\mathbf{b}_t$ of the states.

3.5 Assumptions

We need now to introduce some assumptions that should hold in our setting:

Assumption 3.1. *The smallest element of the transition matrix* $\epsilon := \min_{i,j \in S} \boldsymbol{P}_{i,j} > 0$.

Assumption 3.2. *The reference matrix* $\boldsymbol{A} \in \mathbb{R}^{I^2 V^2 \times S^2}$ *is full column rank.*

Basically, the first assumption gives a non-null probability of transitioning from any state to any other. It is needed for two main reasons. The former is that this assumption implies the regularity of the Chain and, consequently, the presence of a unique stationary distribution, as shown in Proposition 3.1, the latter is mainly a theoretical reason as in our regret analysis we use a result from De Castro et al. (2017) that builds on this condition.
The second assumption, instead, guarantees that the joint distribution of pairs of rewards and pairs of actions given a specific state transition is not the result of a linear combination of the distributions over other state transitions. In the following, we will show that this is a sufficient condition to recover the matrix $\boldsymbol{P}$ since it makes all state transitions distinguishable from the joint pairs of rewards and actions, and it also implies that $I^2 V^2 \geq S^2$.

4 Proposed Approach

4.1 Markov Chain Estimation

As previously stated, the objective is to learn the transition matrix $\boldsymbol{P}$ using the observations we get from the different pulled arms assuming to know the tensor $\mathbf{O} \in \mathbb{R}^{S \times I \times V}$. First of all, we start with a consideration about the transition matrix that defines the chain. Building on Assumption 3.1 and following Proposition 3.1, we can say that exists a unique stationary distribution. This distribution can be easily found by solving the equation below:

$$\boldsymbol{\pi} \boldsymbol{P} = \boldsymbol{\pi}.$$

From the stationary distribution $\boldsymbol{\pi}$, we can define the diagonal matrix $\boldsymbol{\Pi} = diag(\boldsymbol{\pi})$ having the values of the stationary distribution along its diagonal, and we can define the matrix $\boldsymbol{W} = \boldsymbol{\Pi} \boldsymbol{P}$ satisfying $\sum_{i,j \in S} W_{i,j} = 1$. We can see matrix $\boldsymbol{W}$ as the transition matrix $\boldsymbol{P}$ where the transition probabilities from each state (reported along the rows of the transition matrix) are scaled by the probability of the state, given by the stationary distribution. Having defined the matrix $\boldsymbol{W}$, we can interpret the element $W_{i,j}$ as the probability of seeing the transition from state s_i to state s_j when the two consecutive pairs of states are sampled from the mixed Chain. We will also refer to $\boldsymbol{W}$ as the stationary transition distribution matrix. Our objective will be to build an estimate $\widehat{\boldsymbol{W}}$ of the $\boldsymbol{W}$ matrix from which we will derive $\widehat{\boldsymbol{P}}$.
Let's now define an exploration policy θ that selects pairs of arms to be played in successive

rounds. We use this policy for T_0 episodes on MABs that switch according to the underlying Markov Chain, and we obtain a sequence $\mathbb{D} = \{(\mathrm{a}_1, \mathrm{r}_1), (\mathrm{a}_2, \mathrm{r}_2), \dots, (\mathrm{a}_{T_0}, \mathrm{r}_{T_0})\}$. This sequence can also be represented by combining non-overlapping pairs of consecutive elements, thus obtaining $\mathit{Pairs}(\mathbb{D}) = \{(\mathrm{a}_1, \mathrm{a}_2, \mathrm{r}_1, \mathrm{r}_2), \dots, (\mathrm{a}_{T_0-1}, \mathrm{a}_{T_0}, \mathrm{r}_{T_0-1}, \mathrm{r}_{T_0})\}$.
We introduce now the vector $\mathbf{n}_{T_0} \in \mathbb{N}^{I^2 V^2}$ that counts the number of occurrences of elements in $\mathit{Pairs}(\mathbb{D})$. More formally, for each cell of the vector $\mathbf{n}_{T_0}$, we have:

$$\mathrm{n}_{T_0}(a_i, a_j, r_i, r_j) = \sum_{t=0}^{T_0/2} \mathbb{1}\{I_{2t} = a_i, I_{2t+1} = a_j, R_{2t} = r_i, R_{2t+1} = r_j\}.$$

Given the previous considerations, we are now ready to state a core result that links the stationary transition distribution matrix $\boldsymbol{W}$ and the count vector $\mathbf{n}_{T_0}$ as follows:

$$\mathbb{E}[\mathrm{n}_{T_0}(a_i, a_j, r_i, r_j)] = \sum_{s_i, s_j} W_{s_i, s_j} \sum_{t=0}^{T_0/2} \theta(I_{2t} = a_i, I_{2t+1} = a_j) P((R_{2t} = r_i, R_{2t+1} = r_j)|(a_i, a_j), (s_i, s_j)). \quad (5)$$

This equation basically states that a specific couple of rewards will be observed after having pulled a specific couple of arms a number of times which depends on the conditional probabilities of rewards given the couple of arms and each couple of states, weighted by the probability W_{s_i, s_j} that each state transition occurs. We can write the previous formulation in matrix form as follows:

$$\mathbb{E}[\mathbf{n}_{T_0}] = \frac{T_0}{2} \boldsymbol{D} \boldsymbol{A} \boldsymbol{w}, \quad (6)$$

where the matrix $\boldsymbol{A}$ is the reference matrix already defined in Section 3.3, vector $\boldsymbol{w} = \mathit{Vec}(\boldsymbol{W})$ is the vectorization of the matrix $\boldsymbol{W}$, while $\boldsymbol{D} \in \mathbb{R}^{I^2 V^2}$ is a diagonal matrix containing the probabilities (determined by policy θ) associated to each combination of arms, each appearing with multiplicity V^2.
Having defined Equation 6, we are able to compute an estimate of the vector $\widehat{\boldsymbol{w}}$ based on the obtained vector count $\mathbf{n}_{T_0}$:

$$\widehat{\boldsymbol{w}} = \boldsymbol{A}^{\dagger} \widehat{\boldsymbol{D}}_{T_0}^{-1} \mathbf{n}_{T_0}, \quad (7)$$

where $\boldsymbol{A}^{\dagger}$ is the Moore–Penrose inverse of reference matrix $\boldsymbol{A}$ and matrix $\widehat{\boldsymbol{D}}_{T_0}$ is the diagonal matrix that counts with multiplicity V^2 the number of occurrences of each combination of arms (we assume that each combination of arms has been pulled at least once, so $\widehat{\boldsymbol{D}}_{T_0}$ is invertible).
In the limit of infinite samples, Equation 7 has a fixed exact solution that is $\widehat{\boldsymbol{w}} = \boldsymbol{w}$. After the computation of $\widehat{\boldsymbol{w}}$, we obtain an estimate of $\widehat{\boldsymbol{P}}$. The derivation implies two main steps: the first is to write back the vector $\widehat{\boldsymbol{w}}$ in matrix form, reversing the vectorization operation and obtaining matrix $\widehat{\boldsymbol{W}}$; the second step consists in normalizing each obtained row so that the values on each row sum to 1, thus deriving $\widehat{\boldsymbol{P}}$.

4.2 SL-EC Algorithm

Having established an estimation procedure for the transition matrix $\widehat{\boldsymbol{P}}$, we will now provide an algorithm that makes use of this approach in a regret minimization framework.
We consider a finite horizon T for our problem. We propose an algorithm called *Switching Latent Explore then Commit* (SL-EC) that proceeds using an EC approach where the exploration phase is devoted to finding the best estimation of the transition matrix $\widehat{\boldsymbol{P}}$, while during the exploitation phase, we maximize the instantaneous expected reward using the information contained in the belief state $\mathbf{b}$ with the formulation provided in Equation 4. The Exploration phase lasts for T_0 episodes, where T_0 is optimized w.r.t. the total horizon T, as will be seen in Equation 10.
The presented approach is explained in the pseudocode of Algorithm 1.
Basically, a set of all the ordered combinations of pairs of arms is generated at the beginning of the exploration phase, and the pairs of arms are sequentially pulled in a round-robin fashion until the exploration phase is over. The choice of a round-robin approach allows the highlighting of some interesting properties in the

Algorithm 1: SL-EC Algorithm

Input: Reference Matrix $\boldsymbol{A}$, Exploration horizon T_0, Total horizon T
Initialize vector of counts $\mathbf{n} \in \mathbb{N}^{I^2 V^2}$ with zeroes
$t \leftarrow 0$
$\mathbb{D} \leftarrow \{\}$
while $t \leq T_0$ **do**
  foreach $(a_i, a_j) \in I^2$ **do**
    Pull arm $I_t = a_i$
    Observe reward r_t
    Pull arm $I_{t+1} = a_j$
    Observe reward r_{t+1}
    Update $\mathbf{n}$ with $(I_t, I_{t+1}, \mathrm{r}_t, \mathrm{r}_{t+1})$
    $\mathbb{D}.add((I_t, \mathrm{r}_t), (I_{t+1}, , \mathrm{r}_{t+1}))$
    $t \leftarrow t + 2$
$\widehat{\boldsymbol{w}} \leftarrow$ Use Equation 7
$\widehat{\boldsymbol{P}} \leftarrow$ Compute Transition Matrix($\widehat{\boldsymbol{w}}$)
$t \leftarrow 0$
$\mathbf{b}_0 \leftarrow Uniform()$
while $t \leq T$ **do**
  if $t \leq T_0$ **then**
    $I_t = \mathbb{D}.getAction(t)$
  else
    $I_t = \arg\max_{a \in \mathbb{I}} \sum_{s \in \mathbb{S}} \sum_{r \in \mathbb{V}} rQ(r|s,a)\mathrm{b}_{s,t}$
  Observe reward r_t
  $\mathbf{b}_{t+1} \leftarrow UpdateBelief(\mathbf{b}_t, I_t, \mathrm{r}_t)$
  $t \leftarrow t + 1$

theoretical analysis, as will be shown later in Section 5. When the exploration phase is over, an estimation of the transition matrix $\widehat{\boldsymbol{P}}$ is computed using the procedure described in Section 4.1. After that, a belief vector $\mathbf{b}$ is initialized, assigning a uniform probability to all the states, and it is updated using the estimated $\widehat{\boldsymbol{P}}$, considering the history of samples collected during the exploration phase up to T_0. Finally, the exploitation phase starts, as described in the pseudocode of the algorithm.

4.3 Arm Selection Policy

In Algorithm 1, we propose a simple approach for choosing the arms to pull. Each ordered combination of pairs of arms is indeed pulled the same number of times during the exploration phase by using a deterministic approach. However, the estimation framework proposed in Section 4.1 allows for a more flexible arm selection policy. We may randomize the arm choice by assigning non-uniform probabilities to each combination. This aspect allows exploiting the knowledge of the known reward distribution of each arm, for example, giving a higher probability to the combinations of arms that are more rewarding (assuming an initial uniform distribution over state transitions). This arm selection policy may be particularly efficient if we plug this estimation framework into an iterative two-phase exploration and exploitation algorithm, as that used in Zhou et al. (2021). Notably, we could use the estimates of the transition matrix $\widehat{\boldsymbol{P}}_k$ at the end of the k-th exploration phase to properly modify the exploration policy in phase $k+1$ by giving higher probabilities to combinations of arms that are expected to be more rewarding. Indeed, our approach is able to reuse all samples collected during previous exploration phases despite being drawn using different exploration policies.

Offline arm selection In problems with a large number of available arms, a round-robin approach among all possible combinations of pairs may be detrimental as it needs a longer exploration horizon to properly fill the vector count $\mathbf{n}$ and to have better estimation results.

A more convenient approach, in this case, would be to select a subset of different arms, thus leading to a limited number of combinations of pairs of arms to use during the exploration phase. Clearly, in the general case, the removal of some arms may lead to a loss of the total information available. This is not the case when for example we remove a redundant arm, that is an arm that induces the same reward distribution as another arm, given all the latent states. Intuitively, the arm selection procedure tends to promote diversity among arms and remove redundant or similar ones. It turns out we are able to get an understanding of the information loss we suffer by selecting specific arms, given the knowledge of the reference matrix $\boldsymbol{A}$, that we are indeed able to compute beforehand. In particular, in Section 5 devoted to the theoretical analysis, we will see that the expression $\frac{1}{\sigma_{\min}(\boldsymbol{A})}$, with $\sigma_{\min}(\boldsymbol{A})$ representing the minimum singular value of the reference matrix $\boldsymbol{A}$, is an index of the complexity of the problem and we can use this value to drive the choice of the best subset of arms to use. In particular, by fixing a number $J < I$ of arms to use among those available, the choice over the best subset of size J can be done as follows. For each possible subset of arms of size J, we can derive a new reference matrix $\boldsymbol{G}$ from $\boldsymbol{A}$, by extracting from the reference matrix the rows associated with arms' combinations that are feasible using the new subset of arms. At this point, a good candidate subset of arms will be the one with the lowest $\frac{1}{\sigma_{\min}(\boldsymbol{G})}$.
Understandably, this approach implies that the new reference matrix $\boldsymbol{G}$ derived from the subset of selected arms should be full-column rank, thus satisfying Assumption 3.2. It follows that the necessary condition $J^2V^2 \geq S^2$ should be verified.

5 Theoretical Analysis

We will now provide theoretical guarantees on the matrix estimation procedure presented in Section 4.1 and we will prove a regret bound for the SL-EC Algorithm.
We start with a concentration bound on the transition matrix $\widehat{\boldsymbol{P}}$ estimated using samples coming from a round-robin collection policy.

Lemma 5.1. *Suppose Assumptions 3.1 and 3.2 hold. By fixing an exploration parameter T_0 and by pulling each combination of pairs of arms in a round-robin fashion, with probability $1-\delta$ the estimation error of the transition matrix $\boldsymbol{P}$ will be:*

$$\|\boldsymbol{P} - \widehat{\boldsymbol{P}}\|_F \leq \frac{2I^2V}{\sigma_{\min}(\boldsymbol{A})\pi_{\min}} \sqrt{\frac{2S \log \frac{2I^2V^2}{\delta}}{(1-\lambda^{2I^2})T_0}}, \tag{8}$$

where $\|\cdot\|_F$ represents the Frobenius norm (Golub & Van Loan, 1996), $\sigma_{\min}$ represents the minimum singular value of the reference matrix $\boldsymbol{A}$, $\pi_{\min}$ is the minimum component in the probability vector representing the stationary distribution of the Chain, and λ represents the second highest eigenvalue of matrix $\boldsymbol{P}$. We will provide here a sketch of the proof of the presented Lemma. A more detailed version of this proof is reported in Appendix B.

Sketch of the proof The proof of Lemma 5.1 builds on two principal results. The former comprises a relation that links the estimation error of the matrix $\boldsymbol{P}$ with the estimation error of the stationary transition distribution matrix $\boldsymbol{W}$, while the latter is a concentration bound on the estimated $\widehat{\boldsymbol{W}}$ from the true one $\boldsymbol{W}$. Concerning the first result, we can say that:

$$\|\boldsymbol{P} - \widehat{\boldsymbol{P}}\|_F \leq \frac{2\sqrt{S}\|\boldsymbol{W} - \widehat{\boldsymbol{W}}\|_F}{\pi_{\min}}.$$

This result follows from a sequence of algebraic manipulations, also involving a derivation from (Ramponi et al., 2020).
We now need to define a bound on $\|\boldsymbol{W} - \widehat{\boldsymbol{W}}\|_F$. In order to bound this quantity, we apply the vectorization operator $Vec(\cdot)$ to the two matrices obtaining respectively $\boldsymbol{w}$ and $\widehat{\boldsymbol{w}}$ and use the fact that $\|\boldsymbol{W} - \widehat{\boldsymbol{W}}\|_F =$

$\|\boldsymbol{w}-\widehat{\boldsymbol{w}}\|_2$. We proceed as follows:

$$\begin{aligned}\|\boldsymbol{w}-\widehat{\boldsymbol{w}}_{T_0}\|_2 =& \left\|\frac{2}{T_0}A^\dagger\boldsymbol{D}^{-1}(\mathbf{n}_{T_0}-\widehat{\mathbf{n}}_{T_0})\right\|_2 = \left\|\boldsymbol{A}^\dagger(\mathbf{z}-\widehat{\mathbf{z}})\right\|_2 \\ \leq& \|\boldsymbol{A}^\dagger\|_2\|\mathbf{z}-\widehat{\mathbf{z}}\|_2 = \frac{1}{\sigma_{\min}(\boldsymbol{A})}\|\mathbf{z}-\widehat{\mathbf{z}}\|_2,\end{aligned}$$

where in the second equality we replaced the term $(2/T_0)\boldsymbol{D}^{-1}\mathbf{n}_{T_0}$ with the vector $\mathbf{z}\in\mathbb{R}^{I^2V^2}$ and similarly for $\widehat{\mathbf{z}}$. In the inequality instead, we used the consistency property for the spectral norm of matrix $\boldsymbol{A}^\dagger$. Finally, we bound the remaining part as follows:

$$\begin{aligned}\|\mathbf{z}-\widehat{\mathbf{z}}\|_2 =& \sqrt{\sum_{i=1}^{I^2V^2}|\mathrm{z}_i-\widehat{\mathrm{z}}_i|^2} \leq \sqrt{\sum_{i=1}^{I^2V^2}\frac{(1+\lambda^{2I^2})\log\frac{2I^2V^2}{\delta}}{2(1-\lambda^{2I^2})\frac{T_0}{2I^2}}} \\ \leq& \sqrt{\frac{I^2V^2(1+\lambda^{2I^2})\log\frac{2I^2V^2}{\delta}}{2(1-\lambda^{2I^2})\frac{T_0}{2I^2}}} \leq \frac{I^2V}{\sigma_{\min}(\boldsymbol{A})}\sqrt{\frac{2\log\frac{2I^2V^2}{\delta}}{(1-\lambda^{2I^2})T_0}},\end{aligned}$$

where, on the first inequality, we used Hoeffding's inequality with probability $1-\frac{\delta}{I^2V^2}$ for each component of the vector $\widehat{\mathbf{z}}$ and a union bound in the second inequality. In our case, in which samples are generated from a Markov Process, we employed a variant of Hoeffding's inequality that accounts for non-independent samples. We utilized the formulation presented in Fan et al. (2021) which incorporates an additional term $\frac{1+\lambda}{1-\lambda}$ in the bound. More details on this can be found in Proposition C.2 in Appendix C. It is important to note that this proposition holds when the starting distribution of the chain corresponds to the stationary distribution $\boldsymbol{\mu}_0=\boldsymbol{\pi}$, an assumption we can make in our problem. However, if this is not the case, we would suffer a further logarithmic term in the regret (See Theorem 12 in Fan et al. (2021)).
We were able to improve this result by introducing an exponential term $2I^2$ to the second highest eigenvalue λ. This is possible thanks to the adoption of a round-robin procedure for the choice of combinations of arms. Notably, each combination is pulled every $2I^2$ steps of the Markov Process, resulting in a faster mixing of the chain. A more formal result of this aspect can be found in Corollary C.1 in Appendix C.

Having established the results on the estimation matrix $\boldsymbol{P}$, we can now provide regret guarantees for Algorithm 1. The oracle we use is aware of both the observation tensor $\mathbf{O}$ and the transition matrix $\boldsymbol{P}$ but does not observe the hidden state. As well as our algorithm, it builds a belief over the states, using the formulation defined in Equation 3 and selects the arm maximizing the expected instantaneous reward. The derived regret upper bound is provided in the following:

Theorem 5.1. *Suppose Assumptions 3.1 and 3.2 hold. By considering a finite horizon T, there exists a constant T_0, with $T>T_0$, such that with probability $1-\delta$, the regret of the SL-EC Algorithm satisfies:*

$$\mathfrak{R}(T)\leq 2\left(\frac{LI^2V}{\pi_{\min}\sigma_{\min}(\boldsymbol{A})}\sqrt{\frac{2S\log\frac{2I^2V^2}{\delta}}{1-\lambda^{2I^2}}}\cdot T\right)^{2/3}, \tag{9}$$

where L is a constant that depends on the ϵ value appearing in Assumption 3.1 (More details in Appendix C). The presented regret has an order of $\mathcal{O}(T^{2/3})$ w.r.t the horizon T, as common when using an Explore-Then-Commit algorithm. A detailed proof of this theorem can be found in Appendix B. The presented bound on the regret can be achieved by appropriately choosing the exploration horizon T_0. More specifically, we set it as follows:

$$T_0=\left(\frac{LTI^2V}{\sigma_{\min}(\boldsymbol{A})\pi_{\min}}\sqrt{\frac{2S\log\frac{2I^2V^2}{\delta}}{(1-\lambda^{2I^2})}}\right)^{2/3}. \tag{10}$$

5.1 Dependency on the Problem Parameters

By analyzing the results on the bound of the regret, we can observe that it scales with I^2V. This may seem concerning especially when dealing with problems involving a high number of arms or an extremely large number of observations. In particular, this configuration does not allow handling cases with continuous reward models as the number of observations would be infinite, hence impeding the construction of the reference matrix. Fortunately, we can address both aspects, the one related with the dependency on the number of arms and the other on the dependency on the number of observations.

5.1.1 Continuous Reward Distributions and Dependency on the Number of Observations

Concerning the number of observations, it appears that handling continuous reward distributions within this framework is not feasible and this is true if we apply our framework as is. However, nothing prevents us from discretizing the distribution and considering the discretized distribution as a categorical one. The process of discretization involves dividing the distributions into a predetermined number U of distinct segments. Each segment is assigned a probability value that represents the likelihood of a particular sample originating from the continuous distribution and belonging to that segment. Consequently, the count vector is constructed with dimensions I^2U^2, and at each iteration, the associated value of the segment to which the sample belongs is incremented.

The discretization of a continuous distribution paves the way for important considerations because the number of different segments U determines the size of the reference matrix $\boldsymbol{A}$. In principle, we can choose U such that $U^2 \geq S^2$ and this allows us to estimate the transition matrix by using a unique combination of arms (as long as Assumption 3.2 is satisfied). Notably, in the case of continuous distributions and by properly choosing the number of segments, we may need fewer arms to carry on the estimation procedure. It is an interesting problem to determine in this setting the number of suitable splits and the location of the split points that lead to a faster estimation of the transition matrix.

Another issue arises when the environment comprises numerous but finite observations. In such scenarios, we can employ the inverse approach by clustering some observations, thereby reducing the problem's scale. By selecting a number of clusters $C < V$, we can divide the observations into distinct groups. This allows us to utilize cluster-level probabilities (obtained by summing probabilities of the single observations) to construct a new reference matrix and consider counts at the cluster-level for the count vector $\mathbf{n}$. Of course, this approach may lead to a loss of information due to the clustering procedure but it may be beneficial in scenarios with limited availability of memory resources.

5.1.2 Dependency on the Number of Arms

From the point of view of the number of arms, we already observed in Section 4.3 that when the number of arms is large, it is possible to select a subset of arms that allows solving the problem. In particular, the best subset $\mathbb{J}$ we can select is the one minimizing the term $\frac{J^2}{\sigma_{\min}(\boldsymbol{G}_{\mathbb{J}})}$, with J being the size of $\mathbb{J}$ and $\boldsymbol{G}_{\mathbb{J}}$ being the matrix obtained from the choice of the arms in $\mathbb{J}$. It is indeed likely that when $I \gg S$, some arms contain redundant information and can be easily discarded for the estimation procedure.

6 Numerical Simulations

In this section, we provide numerical simulations on synthetic data, demonstrating the effectiveness of the proposed Markov Chain estimation procedure. Specifically, we show the efficiency of the offline arm selection procedure described in Section 4.3 and conduct a comparison between our SL-EC Algorithm and several baselines in non-stationary settings. In Appendix A, we provide additional experiments that highlight the performance difference between our approach and a modified technique based on Spectral Decomposition methods.

Estimation Error of Transition Matrix The first set of experiments is devoted to showing the error incurred by the estimation procedure of the transition matrix in relation to the number of samples considered and the set of actions used for estimation. The left side of Figure 1 illustrates the estimation error of the transition matrix given different instances of Switching Bandits with increasing number of states. In particular, we fix the number of total actions $I = 10$ and number of observations $V = 10$ and consider three instances with $S = 5$, $S = 10$ and $S = 15$ number of states. As it is expected, we can see that as the number

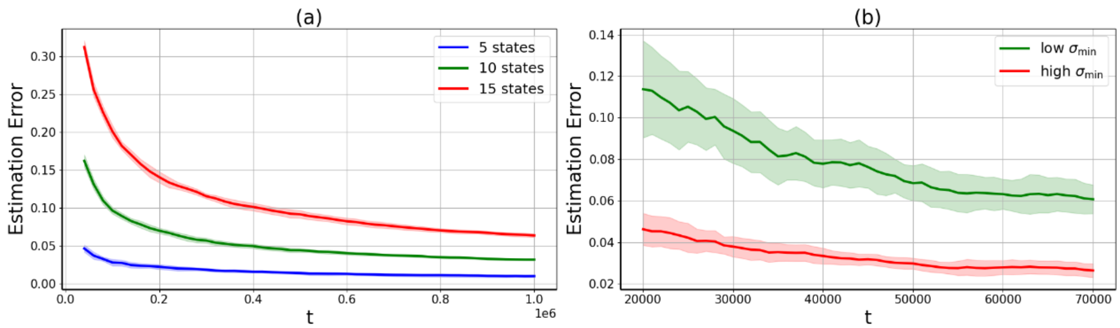

Figure 1: (a) Difference between the estimated and real transition matrix with an increasing number of samples. Metric used is $\|\cdot\|_1$ divided by the number of states (10 runs, 95% c.i.), (b) Difference between real and estimated transition matrix using two different subsets of arms of size $J = 3$ arms from the 8 available on a problem with 5 states. Metric used is $\|\cdot\|_1$ divided by the number of states (10 runs, 95% c.i.).

of states increases the problem becomes more complex, and more samples are needed in order to improve the estimation. Figure 1 reports the $\|\cdot\|_1$ of the error between the true and the estimated transition matrix, scaled by the number of states. We can see that the estimation procedure is particularly efficient leading to low error values even with a limited number of samples, as can be seen from the steep error drop experienced in the first part of the plot.

The right plot in Figure 1, instead, shows the estimation error obtained by using a different subset of arms. As mentioned in previous sections, it is not always beneficial to use all the available actions during the estimation procedure, but selecting a subset of actions may be preferable. Furthermore, we show that by selecting specific subsets of arms we can improve the estimation w.r.t using other subsets. For this experiment, we consider $J = 3$ arms among the $I = 8$ available for a Switching MAB instance with $S = 5$ states. We then identify the optimal subset of arms of size J and initiate the estimation process using the selected subset. In order to find the best one, we generate all matrices of type $\boldsymbol{G}$, as described in Section 4.3 and choose the matrix with lowest $\frac{1}{\sigma_{\min}(\boldsymbol{G})}$. The subset of arms generating that matrix will be used for estimation. The estimation error of the best subset of arms is represented in the plot with the red line, while we represent in green the estimation error of the subset having the lowest $\sigma_{\min}(\boldsymbol{G})$. The figure clearly exhibits the performance difference between the two choices, thereby validating our claims. Additional details about the characteristics of the matrices used in the experiments are provided in Appendix A.

Algorithms Comparisons In this second set of experiments, we compare the regret suffered by our SL-EC approach with other algorithms specifically designed for non-stationary environments. Following the recent work of Zhou et al. (2021), we consider the subsequent baseline algorithms: the simple *ϵ-greedy* heuristics, a sliding-window algorithm such as *SW-UCB* (Garivier & Moulines, 2011) that is generally able to deal with non-stationary settings and the *Exp3.S* (Auer et al., 2002) algorithm. The parameters for all the baseline algorithms have been properly tuned according to the different considered settings. It is worth noting that unlike our Algorithm, the baseline algorithms do not have knowledge of the observation tensor or the underlying Markov Chain. In contrast, our approach utilizes the observation tensor to estimate the transition matrix and to update the belief over the current state. Additionally, we compare our approach with a particle filter algorithm proposed in Hong et al. (2020b) about non-stationary Latent Bandits. They consider two settings: one with complete knowledge of both the observation and transition models and another that incorporates priors on the parameters of the models to account for uncertainty. We compare against a mixture of these two settings by providing their algorithm with full information about the observation model (as it is for our case) and an informative prior about the true transition model.

The comparison is made in terms of the empirical cumulative regret $\widehat{\mathfrak{R}}(t)$, which is the empirical counterpart of the expected cumulative regret $\mathfrak{R}(t)$ averaged over multiple independent runs. These experiments have been conducted on various problem configurations with different numbers of states S, actions I, and observations V. The regret results for these cases are shown in Figure 2. Both plots exhibit similar patterns, with

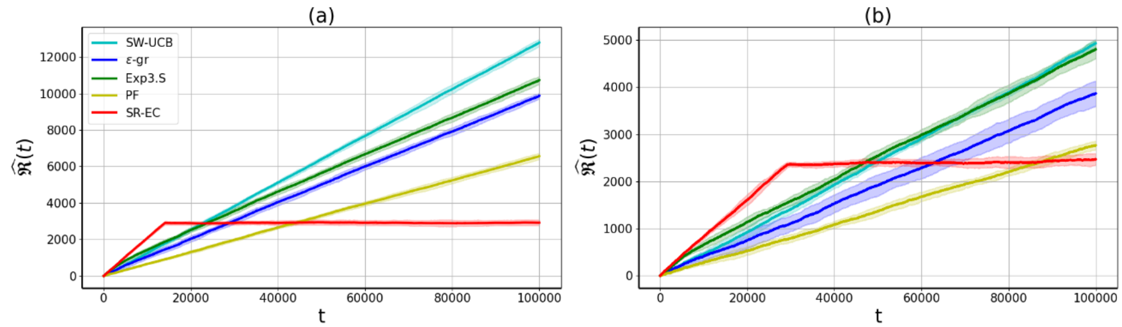

Figure 2: Plots of regret comparing the SL-EC Algorithm with some non-stationary bandit algorithms with a different number of problem parameters: (a) $S = 3$ states, $I = 4$ actions and $V = 5$ observations (5 runs, 95% c.i.); (b) $S = 8$ states, $I = 5$ actions and $V = 10$ observations. (5 runs, 95% c.i.).

most of the baseline algorithms displaying a linear time dependence. This is expected since these algorithms do not take into account the underlying Markov Chain that governs the process. The particle filter algorithm, despite being given a good initial prior on the transition model, is unable to achieve the performance of SL-EC in the long run. Conversely, we can notice a quite different behavior for our algorithm that, in line with an Explore-Then-Commit approach, initially accumulates a large regret and then experiences a drastic slope change when the exploitation phase begins. The regret shown in each plot is the average over all the runs. For further information regarding the generation of the transition model and observation tensor, as well as the hyperparameters used for the baseline algorithms, please refer to Appendix A.
As a remark, our algorithm outperforms the others when the spectral gap β of the chain is not close to zero. Indeed, if this is not the case, simple exploration heuristics such as ϵ-greedy would lead to comparable performance. A clear example is when the transition matrix $\boldsymbol{P}$ defining the chain assigns equal probability to all transitions. In this scenario, all states can be considered independent and identically distributed, and we get no advantage from the knowledge of the matrix $\boldsymbol{P}$ over the use of an algorithm such as ϵ-greedy.

7 Discussion and Conclusions

This paper studies a Latent Bandit problem with latent states changing in time according to an underlying unknown Markov Process. Each state is represented by a different Bandit instance that is unobserved by the agent. As common in the latent Bandit literature, we assumed to know the observation tensor relating each MAB to the reward distribution of its actions, and by using some mild assumptions, we presented a novel estimation technique using the information derived from consecutive pulls of pairs of arms. As far as we know, we are the first to present an estimation procedure of this type aiming at directly estimating the probabilities of the state transitions encoded in the matrix $\boldsymbol{W}$. We have shown that our approach is flexible as it allows choosing combinations of pairs of arms with non-uniform probability and easy as it does not require specific hyperparameters to be set. We also provided some offline techniques for the selection of the best subsets of arms to speed up the estimation process. We analyzed the dependence of the parameters on the complexity of the problem and we showed how our approach can be extended to handle models with continuous observation distributions. We used the presented estimation approach in our SL-EC algorithm that uses an Explore-Then-Commit approach and for which we proved a $\mathcal{O}(T^{2/3})$ regret bound. The experimental evaluation confirmed our theoretical findings showing advantages over some baseline algorithms designed for non-stationary MABs and showing good estimation performances even in scenarios with larger problems.
Some future research directions consists of designing new algorithms that are able to exploit the flexibility in the exploration policy determined by the defined procedure, allegedly in an optimistic way. Furthermore, it may be interesting to deepen the understanding of this problem when dealing with continuous reward models, trying to design optimal ways to discretize them in order to reach faster estimation performances.

References

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15(1):2773–2832, jan 2014.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 138–158, 2019.

Kamyar Azizzadenesheli, Alessandro Lazaric, and Anima Anandkumar. Reinforcement learning of pomdps using spectral methods. In *Annual Conference Computational Learning Theory*, 2016.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, 2014.

Yang Cao, Zheng Wen, Branislav Kveton, and Yao Xie. Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Yohann De Castro, Élisabeth Gassiat, and Sylvain Le Corff. Consistent estimation of the filtering and marginal smoothing distributions in nonparametric hidden markov models. *IEEE Transactions on Information Theory*, 63(8):4758–4777, 2017. doi: 10.1109/TIT.2017.2696959.

Jianqing Fan, Bai Jiang, and Qiang Sun. Hoeffding's inequality for general markov chains and its applications to statistical learning. *Journal of Machine Learning Research*, 22(139):1–35, 2021.

William Feller. *An Introduction to Probability Theory and its Applications Vol. I.* Wiley, 1968.

Tanner Fiez, Shreyas Sekar, and Lillian J. Ratliff. Multi-armed bandits for correlated markovian environments with smoothed reward feedback. *arXiv: Learning*, 2018.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *Proceedings of the 22nd International Conference on Algorithmic Learning Theory*, pp. 174–188, 2011.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, and Craig Boutilier. Latent bandits revisited. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13423–13433. Curran Associates, Inc., 2020a. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/9b7c8d13e4b2f08895fb7bcead930b46-Paper.pdf`.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, Mohammad Ghavamzadeh, and Craig Boutilier. Non-stationary latent bandits. *CoRR*, abs/2012.00386, 2020b.

Fang Liu, Joohyung Lee, and Ness B. Shroff. A change-detection based framework for piecewise-stationary multi-armed bandit problem. In *AAAI Conference on Artificial Intelligence*, 2017.

Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. *31st International Conference on Machine Learning, ICML 2014*, 1, 05 2014.

Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret bounds for restless markov bandits. *Theoretical Computer Science*, 558:62–76, 2014. ISSN 0304-3975. Algorithmic Learning Theory.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley; Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

Giorgia Ramponi, Amarildo Likmeta, Alberto Maria Metelli, Andrea Tirinzoni, and Marcello Restelli. Truly batch model-free inverse reinforcement learning about multiple intentions. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, 2020.

Laurent Saloff-Coste. *Lectures on finite Markov chains*, pp. 301–413. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. ISBN 978-3-540-69210-2.

Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits. In *Annual Conference Computational Learning Theory*, 2008.

Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. Sliding-window thompson sampling for non-stationary settings. *Journal of Artificial Intelligence Research*, 68:311–364, 05 2020. doi: 10.1613/jair.1.11407.

Li Zhou and Emma Brunskill. Latent contextual bandits and their application to personalized recommendations for new users. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 3646–3653. AAAI Press, 2016. ISBN 9781577357704.

Xiang Zhou, Yi Xiong, Ningyuan Chen, and Xuefeng Gao. Regime switching bandits. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.