Regime Switching Bandits

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study a multi-armed bandit problem where the rewards exhibit regime switching. Specifically, the distributions of the random rewards generated from all arms are modulated by a common underlying state modeled as a finite-state Markov chain. The agent does not observe the underlying state and has to learn the transition matrix and the reward distributions. We propose a learning algorithm for this problem, building on spectral method-of-moments estimations for hidden Markov models, belief error control in partially observable Markov decision processes and upper-confidence-bound methods for online learning. We also establish an upper bound $O(T^{2/3}\sqrt{\log T})$ for the proposed learning algorithm where T is the learning horizon. Finally, we conduct proof-of-concept experiments to illustrate the performance of the learning algorithm.

1 Introduction

The multi-armed bandit (MAB) problem is a popular model for sequential decision making with unknown information: the decision maker makes decisions repeatedly among I different options, or arms. After each decision she receives a random reward having an *unknown* probability distribution that depends on the chosen arm. The objective is to maximize the expected total reward over a finite horizon of T periods. The MAB problem has been extensively studied in various fields and applications including Internet advertising, dynamic pricing, recommender systems, clinical trials and medicine [12, 13, 40]. In the classical MAB problem, it is typically assumed that the random reward of each arm is i.i.d. (independently and identically distributed) over time and independent of the rewards from other arms. However, these assumptions do not necessarily hold in practice [10]. To address the drawback, a growing body of literature studies MAB problems with non-stationary rewards to capture temporal changes in the reward distributions in applications, see e.g. [11, 16, 20].

In this paper, we study a non-stationary MAB model with Markovian regime-switching rewards. We assume that the random rewards associated with all the arms are modulated by a *common unobserved* state (or regime) $\{M_t : t = 1, 2, \ldots\}$ modeled as a finite-state discrete-time Markov chain. This chain makes a transition at each period regardless of which arm is pulled and its transition probabilities are independent of the action chosen. Given $M_t = m$, the reward of arm i is i.i.d., whose distribution is denoted $Q(\cdot|m, i)$. Such structural change of the environment is usually referred to as regime switching in finance [32]. The agent doesn't observe or control the underlying state M_t, and has to learn the transition probability matrix P of $\{M_t\}$ as well as the distribution of reward of each arm $Q(\cdot|m, i)$, based on the observed historical rewards. The goal of the agent is to design a learning policy that decides which arm to pull in each period to minimize the expected regret over T periods.

The regime-switching models are widely used in various industries. For example, in finance, the sudden changes of market environments are usually modeled as hidden Markov chains. In revenue management and marketing, a firm may face a shift in consumer demand due to undetected changes in sentiment or competition. In such cases, when the agents (traders or firms) take actions (trading

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

financial assets with different strategies or setting prices), they need to learn the reward and the underlying state at the same time. Our setup is designed to tackle such problems.

Our Contribution. Our study features novel designs in three regards: we propose a new bandit problem formulation, develop a new learning algorithm, and prove regret bounds with new techniques.

In terms of problem formulation, online learning with unobserved states has attracted some attention recently [8, 19]. We consider the strongest oracle among the studies, who knows P and $Q(\cdot|m, i)$, but doesn't observe the hidden state M_t. The oracle thus faces a partially observable Markov decision process (POMDP) [26] (with a long-run average reward objective). By reformulating it as a Markov decision process (MDP) with a continuous belief space (i.e., a distribution over hidden states), the oracle then solves the optimal policy (mapping belief states to actions) using the Bellman equation. Having sublinear regret benchmarked against the strong oracle, our algorithm has better theoretical performance than others with weaker oracles such as the best fixed arm [19] or memoryless policies (the action only depends on the current observation) [8].

In terms of algorithmic design, we propose a learning algorithm (see Algorithm 2) with two key ingredients. First, it builds on the recent advance on the estimation of the parameters of hidden Markov models (HMMs) using spectral method-of-moments methods [3, 2, 8]. It benefits from the theoretical finite-sample bound of spectral estimators, while the finite-sample guarantees of other alternatives such as maximum likelihood estimators remain an open problem [30]. Second, it builds on the well-known "upper confidence bound" (UCB) method in reinforcement learning [7, 25]. There are two difficulties here as the oracle uses the optimal (belief-based) policy of the POMDP. First, the spectral method can not use the non i.i.d. samples generated from the belief-based policy due to the complex history dependency. Second, the belief of the hidden state is subject to the estimation error. Hence, we divide the horizon into nested exploration and exploitation phases. We use spectral estimators in the exploration phase to gauge the estimation error of P and $Q(\cdot|m, i)$. We use the UCB method to control the regret in the exploitation phase. Different from other learning problems, we re-calibrate the belief at the beginning of each exploitation phase based on the parameters estimated in the most recent exploration phase using previous exploratory samples.

In terms of technical analysis, we establish a regret bound of $O(T^{2/3}\sqrt{\log(T)})$ for our proposed learning algorithm where T is the learning horizon. Our regret analysis draws inspirations from [25, 34] for learning MDPs and undiscounted reinforcement learning problems, but the analysis differs significantly from theirs since there are two main technical challenges in our problem.

First, to control the regret, we need to control of the error of the belief state, which itself is not directly observed and needs to be estimated. This is in stark contrast to learning MDPs [25, 34] with observed states. Specifically, we need to bound the estimation error of belief states by the estimation errors of the model parameters. In addition, since the belief state is un-observable, we also need to bound the error in the belief transition kernel, which measures the distance between the belief transition kernel under the optimistic belief MDP model (from the UCB component of our algorithm) at each episode and the belief transition kernel under the true model. These bounds are not trivial since the transition kernel of the belief state depends on the model parameters in a complex way via Bayesian updating. We overcome the difficulties by building on [18] and a delicate analysis of the belief transition kernel to control the errors in the estimations of belief states and the belief transitions.

Second, to establish regret bound, we need an explicit bound for the span of the bias function (also referred as the relative value function) for the belief MDP which has a continuous state space. Such a bound is often critical in the regret analysis of undiscounted reinforcement learning of continuous MDP, but it is either *taken as an assumption* [36] or proved under Hölder continuity assumptions that do not hold for the belief transitions in our setting [34, 27]. We overcome this challenge and bound the bias span by developing a novel approach, which could be of independent interest for learning continuous-state MDPs. Specifically, we bound the bias span by bounding the Lipschitz module of the bias function for our infinite-horizon undiscounted problem (with a long-run average reward objective). To achieve this, we rely on a non-trivial application of the results in [23] which provide general tools for proving Lipschitz continuity of value functions in *finite-horizon discounted* MDPs. One key step is to bound the Lipschitz module of the belief transitions using the Kantorovich metric, which allows us to establish a bound on the Lipschitz module of the value function of the finite-horizon discounted problem *uniformly over the discounting factors*. Exploiting the connection with the infinite-horizon undiscounted problem then yields an explicit bound on the bias span for our problem. We also mention that our bound on the bias span is unrelated to the diameter of the POMDP

discussed in [8]. The diameter in [8] is only for observation-based policies, not for belief-state based policies we consider. That also explains why we need a new approach to bound the bias span.

Related Work. Studies on non-stationary/switching MAB investigate the problem when the rewards of all arms may change over time [5, 20, 10, 6]. Our formulation can be regarded as non-stationary MAB with a special structure. They consider an even stronger oracle than ours, the best arm in each period. However, the total number of changes or the changing budget have to be sublinear in T to achieve sublinear regret. In our formulation, the total number of transitions of the underlying state is linear in T, and the algorithms proposed in these papers fail even considering our oracle (See Section 5). Other studies focus on linear changing budget with some structure such as seasonality [15], which is not present in this paper.

Our work is also related to the restless Markov bandit problem [21, 35, 41] in which the state of each arm evolves according to independent Markov chains. In contrast, our regime-switching MAB model assumes a *common* underlying Markov chain so that the rewards of all arms are correlated, and the underlying state is unobservable to the agent. In addition, our work is related to MAB studies where rewards of all arms depend on a common unknown parameter or a latent random variable, see, e.g., [4, 22, 28, 31, 33]. Our model differs from them in that the common latent state variable follows a dynamic stochastic process which introduces difficulties in algorithm design and analysis.

Two papers have similar settings to ours. [19] studies MAB problems whose rewards are modulated by an unobserved Markov chain and the transition matrix may depend on the action. However, their oracle is the best fixed arm when defining the regret, which is much weaker than the optimal policy of the POMDP (the performance gap is linear between the two oracles). Therefore, their algorithm is expected to have linear regret when using our oracle. [8] also proposes to use spectral estimators to learn POMDPs. Their oracle is the optimal memoryless policy, i.e., a policy that only depends on the current reward observation instead of using all historical observations to form the belief of the underlying state (the performance gap is generally linear in T between their oracle and ours). Such a policy set allows them to circumvent the introduction of the belief entirely. As a result, the dynamics under the memoryless policy can be viewed as a finite-state (modified) HMM and spectral estimators can be applied. Instead, because we consider the optimal belief-based policy, such reduction is not available. Our algorithm and regret analysis hinge on the interaction between the estimation of the belief and the spectral estimators. Our algorithm needs separate exploration to apply the spectral methods and uses the belief-based policy for exploitation. For the regret analysis, unlike [8], we have to carefully control the belief error and bound the span of the bias function from the optimistic belief MDP in each episode. The comparison of our study with related papers is summarized in Table 1. Note that we only present the regret in terms of T.

Papers	Oracle	Changing Budget	Regret
[5]	Best fixed action	Linear	$\tilde{O}(\sqrt{T})$
[20]	Best action in each period	Finite	$\tilde{O}(\sqrt{T})$
[10]	Best action in each period	Sublinear	$\tilde{O}(T^{2/3})$
[8]	Optimal memoryless policy	Linear	$\tilde{O}(\sqrt{T})$
This paper	Optimal POMDP policy	Linear	$\tilde{O}(T^{2/3})$

Table 1: Comparison of our study with some related literature.

2 MAB with Markovian Regime Switching

2.1 Problem Formulation

Consider the MAB problem with arms $\mathcal{I} := \{1, \dots, I\}$. There is a Markov chain $\{M_t\}$ with states $\mathcal{M} := \{1, 2, \dots, M\}$ and transition probability matrix $P \in \mathbb{R}^{M \times M}$. In period $t = 1, 2, \dots$, if the state of the Markov chain $M_t = m$ and the agent chooses arm $I_t = i$, then the reward in that period is R_t with discrete finite support, and its distribution is denoted by $Q(\cdot|m, i) := \mathbb{P}(R_t \in \cdot | M_t = m, I_t = i)$, with $\mu_{m,i} := \mathbb{E}[R_t | M_t = m, I_t = i]$. We use $\boldsymbol{\mu} := (\mu_{m,i}) \in \mathbb{R}^{M \times I}$ to denote the mean reward matrix. The agent knows M and I, but has no knowledge about the underlying state M_t (also referred to as the regime), the transition matrix P or the reward distribution $Q(\cdot|m, i)$. The goal is to

design a learning policy that is adapted to the filtration generated by the observed rewards to decide which arm to pull in each period to maximize the expected cumulative reward over T periods where T is unknown in advance.

If an oracle knows P, $Q(\cdot|m,i)$ and the underlying state M_t, then the problem becomes trivial as s/he would select $I_t^* = \text{argmax}_{i\in\mathcal{I}}\, \mu_{M_t,i}$ in period t. If we benchmark a learning policy against the oracle, then the regret must be linear in T, because the oracle always observes M_t while the agent cannot predict the transition based on the history. Whenever a transition occurs, there is non-vanishing regret incurred. Since the number of transitions during $[0,T]$ is linear in T, the total regret is of the same order. Since comparing to the oracle knowing M_t is uninformative, we consider a weaker oracle who knows P, $Q(\cdot|m,i)$, *but not* M_t. In this case, the oracle solves a POMDP since the states M_t are unobservable and the optimal policy maps belief states (a distribution over the hidden state) to actions. The total expected reward of the POMDP can be shown to scale linearly in T, and asymptotically the reward per period converges to a constant denoted by ρ^* under the optimal belief-based policy. See Section 2.2.

For a learning policy π, we denote by R_t^π the reward received under the learning policy π (which does not know $P, Q(\cdot|m,i)$ initially) in period t. We follow the literature (see, e.g., [25, 35, 1]) and define its total regret after T periods by

$$\mathcal{R}_T \coloneqq T\rho^* - \sum_{t=1}^{T} R_t^\pi. \tag{1}$$

Our goal is to design a learning algorithm with theoretical guarantees including high probability and expectation bounds (sublinear in T) on the total regret.

Without loss of generality we consider Bernoulli rewards with mean $\mu_{m,i} \in (0,1)$ for all m,i. Hence $\mu_{m,i}$ characterizes the distribution $Q(\cdot|m,i)$. Our analysis holds generally for random rewards with discrete finite support. In addition, we impose the following assumptions.

Assumption 1. *The transition matrix P of the Markov chain $\{M_t\}$ is invertible.*

Assumption 2. *The mean reward matrix $\boldsymbol{\mu} = (\mu_{m,i})$ has full row rank.*

Assumption 3. *The smallest element of the transition matrix $\epsilon \coloneqq \min_{i,j\in\mathcal{M}} P_{ij} > 0$.*

The first two assumptions are required for finite-sample guarantees of spectral estimators for HMMs [3, 2]. The third assumption is needed to control the belief error of the hidden states by the state-of-art methods; See [18] and our Proposition 3. We next reformulate the POMDP as a belief MDP.

2.2 Reduction of POMDP to Belief MDP

To present our learning algorithm and analyze the regret, we first investigate the POMDP problem faced by the oracle where parameters P, $\boldsymbol{\mu}$ (equivalently Q for Bernoulli rewards) are known with unobserved states M_t. Based on the historical observed history, the oracle forms a belief of the underlying state. The belief can be encoded by a M-dimension vector $b_t = (b_t(1),\ldots,b_t(M)) \in \mathcal{B}$: $b_t(m) \coloneqq \mathbb{P}(M_t = m|I_1,\cdots,I_{t-1},R_1,\cdots,R_{t-1})$, where $\mathcal{B} \coloneqq \left\{b \in \mathbb{R}_+^M : \sum_{m=1}^M b(m) = 1\right\}$. It is well known that the POMDP of the oracle can be seen as a MDP built on a (continuous) belief state space $\mathcal{B}$, and here a policy is a mapping from belief states to actions, see [26].

We next introduce a few notations that facilitate the analysis. For notation simplicity, we write $c(m,i) \coloneqq \mu_{m,i}$. Given belief $b \in \mathcal{B}$, the expected reward of arm i is $\bar{c}(b,i) \coloneqq \sum_{m=1}^M c(m,i)b(m)$. In period t, given belief $b_t = b, I_t = i, R_t = r$, by Bayes' theorem, the belief b_{t+1} is updated by $b_{t+1} = H_{\boldsymbol{\mu},P}(b,i,r)$, where the m-th entry is $b_{t+1}(m) = \frac{\sum_{m'} P(m',m)\cdot(\boldsymbol{\mu}_{m',i})^r(1-\boldsymbol{\mu}_{m',i})^{1-r}\cdot b_t(m')}{\sum_{m''}(\boldsymbol{\mu}_{m'',i})^r(1-\boldsymbol{\mu}_{m'',i})^{1-r}\cdot b_t(m'')}$. It is obvious that the forward function H depends on the transition matrix P and the reward matrix $\boldsymbol{\mu}$. We can also define the transition probability of the belief state conditional on the arm pulled: $\bar{T}(\cdot|b,i) \coloneqq \mathbb{P}(b_{t+1} \in \cdot|b,i)$, where b_{t+1} is random due to the random reward.

The long-run average reward of the infinite-horizon belief MDP following policy π given the initial belief b can be written as $\rho_b^\pi \coloneqq \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}[\sum_{t=1}^T R_t^\pi|b_1 = b]$. The optimal (belief-based) policy maximizes ρ_b^π for a given b. We use $\rho^* \coloneqq \sup_b \sup_\pi \rho_b^\pi$ to denote the optimal long-run

average reward. Under this belief MDP formulation, for all $b \in \mathcal{B}$, the Bellman equation states that

$$\rho^* + v(b) = \max_{i \in \mathcal{I}} \left[\bar{c}(b,i) + \int_{\mathcal{B}} \bar{T}(db'|b,i) v(b') \right], \tag{2}$$

where $v : \mathcal{B} \mapsto \mathbb{R}$ is the bias function. It can be shown (see the proof of Proposition 2 in Appendix) that under our assumptions, ρ^* and $v(b)$ are well defined and there exists a stationary deterministic optimal policy π^* which maps a belief state to an arm to pull (an action that maximizes the right side of (2)). Various (approximate) methods have been proposed to solve (2) and to find the optimal policy for the belief MDP, see e.g. [42, 38, 39]. In this work, we do not focus on this planning problem for a known model, and we assume the access to an optimization oracle that solve (2) and returns the optimal average reward ρ^* and the optimal stationary policy for a given known model.

3 The SEEU Algorithm

This section describes our learning algorithm for the regime switching MAB model: the Spectral Exploration and Exploitation with UCB (SEEU) algorithm. To device a learning policy for the POMDP with unknown $\boldsymbol{\mu}$ and P, one needs a procedure to estimate those quantities from observed rewards. [3, 2] propose the so-called spectral estimator for the unknown parameters in HMMs. However, the algorithm is not directly applicable to ours, because there is no decision making in HMMs. To use the spectral estimator, we divide the learning horizon T into nested "exploration" and "exploitation" phases. In the exploration phase, we randomly select an arm in each period. This transforms the system into a HMM so that we can apply the spectral method to estimate $\boldsymbol{\mu}$ and P from the observed rewards in the phase. In the exploitation phase, based on the estimators obtained from the exploration phase, we use a UCB-type policy to further narrow down the optimal belief-based policy in the POMDP introduced in Section 2.2.

3.1 Spectral Estimator

We introduce the spectral estimator [3, 2], and adapt it to our setting. To simplify the notation, suppose the exploration phase starts from period 1 until period n, with realized arms $\{i_1, \ldots, i_n\}$, and realized rewards $\{r_1, \ldots, r_n\}$ sampled from Bernoulli distributions. Recall I is the cardinality of the arm set $\mathcal{I}$, then one can create a one-to-one mapping from a pair (i, r) into a scalar $s \in \{1, 2, ..., 2I\}$. Therefore, the pair can be expressed as a vector $y \in \{0,1\}^{2I}$ such that in each period t, y_t satisfies $\mathbb{1}_{\{y_t = e_s\}} = \mathbb{1}_{\{r_t = r, i_t = i\}}$, where e_s is a basis vector with its s-th element being one and zero otherwise. Let $A \in \mathbb{R}^{2I \times M}$ be the observation probability matrix conditional on the state: $A(s,m) = \mathbb{P}(R_t = r, I_t = i | M_t = m)$. It can be shown that A satisfies $\mathbb{E}[y_t | M_t = m] = Ae_m$, and $\mathbb{E}[y_{t+1} | M_t = m] = AP^T e_m$. Write $\otimes$ for the tensor product. For three consecutive observations y_{t-1}, y_t, y_{t+1}, define

$$\widetilde{y}_{t-1} \coloneqq \mathbb{E}[y_{t+1} \otimes y_t]\mathbb{E}[y_{t-1} \otimes y_t]^{-1} y_{t-1}, \qquad \widetilde{y}_t \coloneqq \mathbb{E}[y_{t+1} \otimes y_{t-1}]\mathbb{E}[y_t \otimes y_{t-1}]^{-1} y_t,$$
$$M_2 \coloneqq \mathbb{E}[\widetilde{y}_{t-1} \otimes \widetilde{y}_t], \qquad M_3 \coloneqq \mathbb{E}[\widetilde{y}_{t-1} \otimes \widetilde{y}_t \otimes y_{t+1}].$$

From the observations $\{y_1, y_2, \ldots, y_n\}$, we may construct the estimations $\hat{M}_2$ and $\hat{M}_3$ for M_2 and M_3 respectively, and apply the tensor decomposition to obtain the estimator $\hat{\boldsymbol{\mu}}$ for the unknown mean reward matrix and $\hat{P}$ for the transition matrix. This procedure is summarized in Algorithm 1. We use $\hat{A}_m$ (respectively $\hat{B}_m$) to denote the m-th column vector of $\hat{A}$ (respectively $\hat{B}$). In addition, we have the following result ([8]) and it provides the confidence regions of the estimators in Algorithm 1.

Proposition 1. *Under Assumptions 1 and 2, for any $\delta \in (0,1)$ and any initial distribution, there exists N_0 such that when $n \geq N_0$, with probability $1 - \delta$, the estimated $\hat{\boldsymbol{\mu}}$ and $\hat{P}$ by Algorithm 1 satisfy*

$$||(\boldsymbol{\mu})^m - (\hat{\boldsymbol{\mu}})^m||_2 \leq C_1 \sqrt{\frac{\log(6\frac{S^2+S}{\delta})}{n}}, \quad m \in \mathcal{M},$$
$$||P - \hat{P}||_2 \leq C_2 \sqrt{\frac{\log(6\frac{S^2+S}{\delta})}{n}}. \tag{3}$$

where $(\boldsymbol{\mu})^m$ and $(\hat{\boldsymbol{\mu}})^m$ are the m-th row vectors of $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$, respectively. Here, $S = 2I$, and C_1, C_2 are constants independent of n.

Algorithm 1 The subroutine to estimate $\boldsymbol{\mu}$ and P from the observations from the exploration phase.

Input: sample size n, $\{y_1, y_2, \ldots, y_n\}$ created from the rewards $\{r_1, \ldots, r_n\}$ and arms $\{i_1, \ldots, i_n\}$

Output: The estimation $\hat{\boldsymbol{\mu}}, \hat{P}$

1: For $i, j \in \{-1, 0, 1\}$: compute $\hat{W}_{i,j} = \frac{1}{N-2} \sum_{t=2}^{N-1} y_{t+i} \otimes y_{t+j}$.

2: For $t = 2, \ldots, n-1$: compute $\hat{y}_{t-1} := \hat{W}_{1,0}(\hat{W}_{-1,0})^{-1} y_{t-1}$, $\hat{y}_t := \hat{W}_{1,-1}(\hat{W}_{0,-1})^{-1} y_t$.

3: Compute $\hat{M}_2 := \frac{1}{N-2} \sum_{t=2}^{N-1} \hat{y}_{t-1} \otimes \hat{y}_t$, $\hat{M}_3 := \frac{1}{N-2} \sum_{t=2}^{N-1} \hat{y}_{t-1} \otimes \hat{y}_t \otimes y_{t+1}$.

4: Apply tensor decomposition ([2]):
$\hat{B} =$ **TensorDecomposition**$(\hat{M}_2, \hat{M}_3)$.

5: Compute $\hat{A}_m = \hat{W}_{-1,0}(\hat{W}_{1,0})^{\dagger} \hat{B}_m$ for each $m \in \mathcal{M}$.

6: Return mth row vector $(\hat{\boldsymbol{\mu}})^m$ of $\hat{\boldsymbol{\mu}}$ from $\hat{A}_m$.

7: Return $\hat{P} = (\hat{A}^{\dagger} \hat{B})^{\top}$ ($\dagger$ represents the pseudoinverse of a matrix)

The expressions of constants N_0, C_1, C_2 are given in Section B in the appendix. Note that parameters $\boldsymbol{\mu}, P$ are identifiable up to permutations of the hidden states [8].

3.2 The SEEU Algorithm

The SEEU algorithm proceeds in episodes of increasing length, similar to UCRL2 in [25]. As mentioned before, each episode is divided into exploration and exploitation phases. In episode k, it starts with the exploration phase that lasts for a fixed number of periods τ_1, and the algorithm uniformly randomly chooses an arm and observes the rewards. After the exploration phase, the algorithm applies Algorithm 1 to (re-)estimate $\boldsymbol{\mu}$ and P. Moreover, it constructs a confidence interval based on Proposition 1 with a confidence level $1 - \delta_k$, where $\delta_k := \delta / k^3$ is a vanishing sequence. Then the algorithm enters the exploitation phase. Its length is proportional to $\sqrt{k}$. In the exploitation phase, it conducts UCB-type learning: the arm is pulled according to a policy that corresponds to the optimistic estimator of $\boldsymbol{\mu}$ and P inside the confidence interval. The detailed steps are listed in Algorithm 2. Note we use the UCB component in the exploitation phase instead of point estimators of $\boldsymbol{\mu}$ and P for the ease of analysis.

Algorithm 2 The SEEU Algorithm

Input: Initial belief b_1, precision δ, exploration parameter τ_1, exploitation parameter τ_2

1: **for** $k = 1, 2, 3, \ldots$ **do**

2: Set the start time of episode k, $t_k := t$

3: **for** $t = t_k, t_k + 1, \ldots, t_k + \tau_1$ **do**

4: Uniformly randomly select an arm: $\mathbb{P}(I_t = i) = \frac{1}{I}$

5: **end for**

6: Input the realized actions and rewards in all previous exploration phases $\hat{\mathcal{I}}_k := \{i_{t_1:t_1+\tau_1}, \cdots, i_{t_k:t_k+\tau_1}\}$ and $\hat{\mathcal{R}}_k := \{r_{t_1:t_1+\tau_1}, \cdots, r_{t_k:t_k+\tau_1}\}$ to Algorithm 1 to compute $\hat{\boldsymbol{\mu}}_k, \hat{P}_k =$ **Spectral Estimation**$(\hat{\mathcal{I}}_k, \hat{\mathcal{R}}_k)$

7: Compute the confidence interval $\mathcal{C}_k(\delta_k)$ from (3) using the confidence level $1 - \delta_k = 1 - \frac{\delta}{k^3}$ such that $\mathbb{P}\{(\boldsymbol{\mu}, P) \in \mathcal{C}_k(\delta_k)\} \geq 1 - \delta_k$

8: Find the optimistic POMDP in the confidence interval
$$(\boldsymbol{\mu}_k, P_k) = \operatorname{argmax}_{(\boldsymbol{\mu}, P) \in \mathcal{C}(\delta_k)} \rho^*(\boldsymbol{\mu}, P)$$

9: **for** $t = 1, 2, \ldots, t_k + \tau_1$ **do**

10: Update belief b_t^k to $b_{t+1}^k = H_{\boldsymbol{\mu}_k, P_k}(b_t^k, i_t, r_t)$ under the new parameters $(\boldsymbol{\mu}_k, P_k)$

11: **end for**

12: **for** $t = t_k + \tau_1 + 1, \ldots, t_k + \tau_2 \sqrt{k}$ **do**

13: Execute the optimal policy $\pi^{(k)}$ by solving the Bellman equation (2) under parameters $(\boldsymbol{\mu}_k, P_k)$: $i_t = \pi^{(k)}(b_t^k)$

14: Observe reward r_t

15: Update the belief at $t + 1$ by $b_{t+1}^k = H_{\boldsymbol{\mu}_k, P_k}(b_t^k, i_t, r_t)$

16: **end for**

17: **end for**

3.3 Discussions on the SEEU Algorithm

Computations. For given parameters $(\boldsymbol{\mu}, P)$, we need to compute the optimal average reward $\rho^*(\boldsymbol{\mu}, P)$ that depends on the parameters (Step 8 in Algorithm 2). Various computational and approximation methods have been proposed to tackle this planning problem for belief MDPs as mentioned in Section 2.2. In addition, we need to find out the optimistic POMDP in the confidence region $\mathcal{C}_k(\delta_k)$ with the best average reward. For low dimensional models, one can discretize $\mathcal{C}_k(\delta_k)$ into grids and calculate the corresponding optimal average reward ρ^* at each grid point so as to find (approximately) the optimistic model $(\boldsymbol{\mu}_k, P_k)$. However, in general it is not clear whether there is an efficient computational method to find the optimistic plausible POMDP model in the confidence region when the unknown parameters are high-dimensional. The extended value iteration method in [25] does not work in our POMDP setting. This is because we cannot separately find $\boldsymbol{\mu}$ and P as $\bar{c}$ and $\bar{T}$ in (2) both depend on $\boldsymbol{\mu}$, and we cannot write the inner optimization in the extended value iteration to find optimistic P as a linear programming over the convex polytope $\mathcal{C}_k(\delta_k)$ since $\bar{T}$ is a nonlinear function of $\boldsymbol{\mu}$ and P. This issue is also present in recent studies on learning continuous-state MDPs with the upper confidence bound approach, see e.g. [27] for further discussions. In our regret analysis below, we do not take into account approximation errors arising from the computational aspects discussed above, as in [34, 27].

Dependence on the unknown parameters. When computing the confidence region in Step 7 of Algorithm 2, the agent needs the information of the constants C_1 and C_2 in Proposition 1. These constants depend on a few "primitives" that can be hard to know, for example, the mixing rate of the underlying Markov chain. However we only need upper bounds for C_1 and C_2 for the theoretical guarantee, and hence a rough and conservative estimate would be sufficient. Such dependence on some unknown parameters is common in learning problems, and one remedy is to dedicate the beginning of the horizon to estimate the unknown parameters, which typically doesn't increase the rate of the regret. Alternatively, C_1 and C_2 can be replaced by parameters that are tuned by hand. See Remark 3 of [8] for a further discussion on this issue.

4 Regret Bound

This section presents our main theoretical results on the regret bound for the SEEU algorithm. We first state two technical results that are important in proving the regret bounds.

Proposition 2 (Uniform bound on the bias span)**.** *If the belief MDP satisfies Assumption 3, then for (ρ, v) satisfying the Bellman equation* (2)*, we have the span of the bias function span$(v) := \max_{b\in\mathcal{B}} v(b) - \min_{b\in\mathcal{B}} v(b)$ is bounded by $D(\epsilon)$, where*

$$D(\epsilon) := \frac{8\left(\frac{2}{(1-\alpha)^2} + (1+\alpha)\log_\alpha \frac{1-\alpha}{8}\right)}{1-\alpha}, \quad \textit{with } \alpha = \frac{1-2\epsilon}{1-\epsilon} \in (0,1).$$

Recall v_k is the bias function for the optimistic belief MDP in episode k. Proposition 2 guarantees that $span(v_k)$ is bounded by $D = D(\epsilon/2)$ uniformly in k, because Assumption 3 can be satisfied (with ϵ replaced by $\epsilon/2$) by the optimistic MDPs when T is sufficiently large due to Proposition 1.

Proposition 3 (Controlling the belief error)**.** *Suppose Assumption 3 holds. Given $(\hat{\boldsymbol{\mu}}, \hat{P})$, an estimator of the true model parameters $(\boldsymbol{\mu}, P)$. For an arbitrary reward-action sequence $\{r_{1:t}, i_{1:t}\}_{t\geq 1}$, let $\hat{b}_t$ and b_t be the corresponding beliefs in period t under $(\hat{\boldsymbol{\mu}}, \hat{P})$ and $(\boldsymbol{\mu}, P)$ respectively. Then there exists constants L_1, L_2 such that*

$$||\hat{b}_t - b_t||_1 \leq L_1||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||_1 + L_2||\hat{P} - P||_F,$$

where $L_1 = 4M(\frac{1-\epsilon}{\epsilon})^2/\min\{\boldsymbol{\mu}_{\min}, 1-\boldsymbol{\mu}_{\max}\}$, $L_2 = 4M(1-\epsilon)^2/\epsilon^3 + \sqrt{M}$, $||\cdot||_F$ is the Frobenius norm, $\boldsymbol{\mu}_{\max}$ and $\boldsymbol{\mu}_{\min}$ are the maximum and minimum element of the matrix $\boldsymbol{\mu}$ respectively.

We now state our first main result. The proof is given in Appendix E.

Theorem 1. *Suppose Assumptions 1 to 3 hold. Fix the parameter τ_1 in Algorithm 2 to be sufficiently large. Then there exist constants T_0, C which are independent of T, such that when $T > T_0$, with probability $1 - \frac{7}{2}\delta$, the regret of Algorithm 2 satisfies*

$$\mathcal{R}_T \leq CT^{2/3}\sqrt{\log\left(\frac{3(S+1)}{\delta}T\right)} + T_0\rho^*,$$

where $S = 2I$ and ρ^* denotes the optimal long-run average reward under the true model.

The constant T_0 measures the number of periods needed for the sample size in the exploration phases to exceed N_0 arising in Proposition 1. The constant C has the following expression:

$$C = 3\sqrt{2}\left[\left(D + 1 + \left(1 + \frac{D}{2}\right)L_1\right)M^{3/2}C_1 + \left(1 + \frac{D}{2}\right)L_2 M^{1/2}C_2\right]\tau_2^{1/3}\tau_1^{-1/2} + 3\tau_2^{-2/3}(\tau_1\rho^* + D) + (D+1)\sqrt{2\ln(\frac{1}{\delta})}.$$

Here, M is the number of hidden states, C_1, C_2 are given in Proposition 1, D is given in Proposition 2, and L_1, L_2 are given in Proposition 3. One can verify $C = \tilde{O}(M^{5/2}I)$, ignoring logarithmic factors. For technical reasons we require τ_1 to be large so as to apply Proposition 1 from the first episode. The impact of the hyper-parameters τ_1 and τ_2 on the regret will be studied numerically in Section 5.

Remark 2 (Lengths of exploration and exploitation phases)**.** *In Algorithm 2, we use an exploration phase of length τ_1 and a exploitation phase of length $\tau_2\sqrt{k}$ in episode k. This in the end leads to $O(T^{2/3})$ of the regret as the total length of exploration (and the total number of episodes) is $\Theta(T^{2/3})$. The intuition of such a choice is the following. After k episodes, the total length of exportation is $\tau_1 k$, and hence the estimation error of the model parameters and also the belief states is of order $1/\sqrt{k}$. As a result, we needs to choose an exploitation phase with length $\tau_2\sqrt{k}$ to balance this so that the regret incurred per episode is controlled. One might wonder whether one can simply choose a longer exploitation phase of length $\tau_2 k^\alpha$ with $\alpha \geq 1/2$, and then the regret order will be reduced. This is not possible, and the intuition is as follows. The total number of episodes K satisfy $\sum_{k=1}^{K}(\tau_1 + \tau_2 k^\alpha) = T$, which yields $K \sim T^{\frac{1}{\alpha+1}}$. The total regret incurred then is, ignoring the logarithmic factors, on the order of $\sum_{k=1}^{K}(\tau_1 + \frac{1}{\sqrt{k}} \cdot \tau_2 k^\alpha)$, which is $T^{\frac{\alpha+1/2}{\alpha+1}}$. One can immediately see that the choice of $\alpha = 1/2$ is actually optimal, leading to a regret of order $T^{2/3}$.*

From Theorem 1, we can choose an appropriate $\delta = \frac{3(S+1)}{T}$ and readily obtain the following expectation bound on the regret. The proof is omitted.

Theorem 3. *Under the same assumptions as Theorem 1, the regret of Algorithm 2 satisfies*

$$\mathbb{E}[\mathcal{R}_T] \leq CT^{2/3}\sqrt{2\log T} + (T_0 + 11(S+1))\rho^*.$$

Remark 4 (Lower bound)**.** *For the lower bound of the regret, consider the I problem instances (equal to the number of arms): In instance i, let $\mu_{m,i} = 0.5 + \bar{\epsilon}$ for all m for a small positive constant $\bar{\epsilon}$, and let $\mu_{m,j} = 0.5$ for all m and $j \neq i$. Such structure makes sure that the oracle policy simply pulls one arm without the need to infer the state. Since the problem reduces to the classic MAB, the regret is at least $O(\sqrt{IT})$ in this case. Note that the setup of the instances may violate Assumption 2, but this can be easily fixed by introducing an arbitrarily small perturbation to $\boldsymbol{\mu}$. The gap between the upper and lower bounds is probably caused by the split of exploration/exploitation phases in our algorithm, which resembles the $O(T^{2/3})$ regret for explore-then-commit algorithms in classic multi-armed bandit problems (see Chapter 6 in [29]). We cannot integrate the two phases because of the following fundamental barrier: the spectral estimator cannot use samples generated from the belief-based policy due to the history dependency. This is also why [8] focused on observation-based policies in order to apply the spectral estimator. Since the spectral estimator is the only method that provides finite-sample guarantees for HMMs, we leave the gap as a future direction of research. Nevertheless, we are not aware of other algorithms that can achieve sublinear regret in our setting.*

Remark 5 (General reward distribution)**.** *Our model requires the estimation of the reward distribution instead of just the mean to estimate the belief. For discrete rewards taking O possible values, the regret bound holds with $S = OI$ instead of $2I$ for Bernoulli rewards. For continuous reward distributions, it might be possible to combine the non-parametric HMM inference method in [18] with our algorithm design to obtain regret bounds, and we leave it for future work.*

5 Numerical Experiment

In this section, we present proof-of-concept experiments. Note that large-scale POMDPs with long-run average objectives (the oracle in our problem) are computationally difficult to solve, i.e.

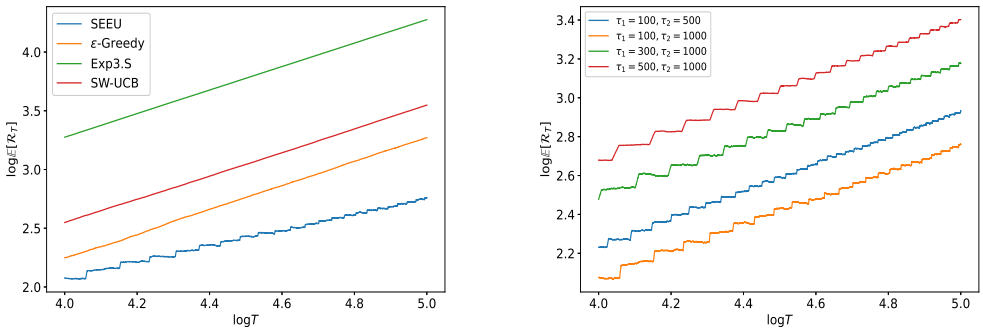

Figure 1: (a) Regret comparison of four algorithms; (b) Effect of (τ_1, τ_2) on the regret.

it is challenging to find the optimal belief-based policy [14]. On the other hand, while there can be many hidden states in general, often only two or three states are important to model in several application areas, e.g. "bull market" and "bear market" in finance [17]. Hence we focus on small-scale experiments, following some recent literature on reinforcement learning for POMDPs [8, 24].

As a representative example, we consider a 2-hidden-state, 2-arm setting with $P = \begin{bmatrix} 1/3 & 2/3 \\ 3/4 & 1/4 \end{bmatrix}$ and $\boldsymbol{\mu} = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.6 \end{bmatrix}$, where the random reward follows a Bernoulli distribution. We compare our algorithm with simple heuristics ϵ-greedy ($\epsilon = 0.1$), and non-stationary bandits algorithms including Sliding-Window UCB (SW-UCB) [20] with tuned window size and Exp3.S [5] with $L = T$, where the hyperparameter L is the number of changes in their algorithm. In Figure 1(a), we plot the average regret versus T of the four algorithms in log-log scale, where the number of runs for each algorithm is 50. The numerical experiments are conducted on a PC with 3.10 GHz Intel Processor and 16 GB of RAM. We observe that the slopes of all algorithms except for SEEU are close to one, suggesting that they incur linear regrets. This is expected, because these algorithms don't take into account the hidden states. On the other hand, the slope of SEEU is close to $2/3$. This is consistent with our theoretical result (Theorem 3). Similar observations are made on other small-scale examples. This demonstrates the effectiveness of our SEEU algorithm, particularly when the horizon length T is relatively large.

We also briefly discuss the impact of parameters τ_1 and τ_2 on the algorithm performance. For the example above, we calculate the average regret for several pairs of parameters (τ_1, τ_2). It can be seen that the choices of these parameters do not affect the order $O(T^{2/3})$ of the regret (the slope). See Figure 1(b) for an illustration.

6 Conclusions, Limitations and Future Research

In this paper, we study a non-stationary MAB model with Markovian regime-switching rewards. We propose a learning algorithm that integrates spectral estimators for hidden Markov models and upper confidence methods from reinforcement learning. We also establish a regret bound of order of $O(T^{2/3}\sqrt{\log T})$ for the learning algorithm. As far as we know, this is the first algorithm with sublinear regret for MAB with unobservable regime switching.

The main limitation of our work is that the regret upper bound does not match the lower bound. It would be interesting to find out whether one can improve the regret upper bound $O(T^{2/3}\sqrt{\log T})$. The gap in the upper and lower bounds are likely to be filled if one can resolve either of the following two fundamental open questions: (1) show the spectral method can be applied to non i.i.d. samples generated from belief-based policies, and (2) establish finite-sample guarantees for other estimators (e.g. maximum likelihood estimators) of POMDP parameters with data generated from adaptive policies. We leave them for future research.

References

[1] S. Agrawal and R. Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *In Advances in Neural Information Processing Systems*, pages 1184–1194, 2017.

[2] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[3] A. Anandkumar, D. Hsu, and S. M. Kakade. A method of moments for mixture models and hidden markov models. In *Conference on Learning Theory*, pages 33–1, 2012.

[4] O. Atan, C. Tekin, and M. van der Schaar. Global bandits. *IEEE transactions on neural networks and learning systems*, 29(12):5798–5811, 2018.

[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[6] P. Auer, P. Gajane, and R. Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, pages 138–158, 2019.

[7] P. Auer and R. Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 49–56, 2007.

[8] K. Azizzadenesheli, A. Lazaric, and A. Anandkumar. Reinforcement learning of POMDPs using spectral methods. *arXiv preprint arXiv:1602.07764*, 2016.

[9] K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.

[10] O. Besbes, Y. Gur, and A. Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in neural information processing systems*, pages 199–207, 2014.

[11] O. Besbes, Y. Gur, and A. Zeevi. Optimal exploration-exploitation in a multi-armed-bandit problem with non-stationary rewards. *Stochastic Systems*, 2019.

[12] D. Bouneffouf and I. Rish. A survey on practical applications of multi-armed and contextual bandits. *arXiv preprint arXiv:1904.10040*, 2019.

[13] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[14] K. Chatterjee, R. Saona, and B. Ziliotto. The complexity of pomdps with long-run average objectives. *arXiv preprint arXiv:1904.13360*, 2019.

[15] N. Chen, C. Wang, and L. Wang. Learning and optimization with seasonal patterns. *Working paper*, 2020.

[16] W. C. Cheung, D. Simchi-Levi, and R. Zhu. Learning to optimize under non-stationarity. *arXiv preprint arXiv:1810.03024*, 2018.

[17] M. Dai, Q. Zhang, and Q. J. Zhu. Trend following trading under a regime switching model. *SIAM Journal on Financial Mathematics*, 1(1):780–810, 2010.

[18] Y. De Castro, E. Gassiat, and S. 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.

[19] T. Fiez, S. Sekar, and L. J. Ratliff. Multi-armed bandits for correlated markovian environments with smoothed reward feedback. *arXiv preprint arXiv:1803.04008*, 2019.

[20] A. Garivier and E. Moulines. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pages 174–188, Berlin, Heidelberg, 2011. Springer.

[21] S. Guha, K. Munagala, and P. Shi. Approximation algorithms for restless bandit problems. *Journal of the ACM (JACM)*, 58(1):3, 2010.

[22] S. Gupta, G. Joshi, and O. Yağan. Correlated multi-armed bandits with a latent random source. *arXiv preprint arXiv:1808.05904*, 2018.

[23] K. Hinderer. Lipschitz continuity of value functions in markovian decision processes. *Mathematical Methods of Operations Research*, 62(1):3–22, 2005.

[24] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson. Deep variational reinforcement learning for pomdps. *arXiv preprint arXiv:1806.02426*, 2018.

[25] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

[26] V. Krishnamurthy. *Partially observed Markov decision processes*. Cambridge University Press, 2016.

[27] K. Lakshmanan, R. Ortner, and D. Ryabko. Improved regret bounds for undiscounted continuous reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 524–532, 2015.

[28] T. Lattimore and R. Munos. Bounded regret for finite-armed structured bandits. In *Advances in Neural Information Processing Systems*, pages 550–558, 2014.

[29] T. Lattimore and C. Szepesvári. *Bandit algorithms*. preprint, 2018.

[30] L. Lehéricy. Consistent order estimation for nonparametric hidden markov models. *Bernoulli*, 25(1):464–498, 2019.

[31] O. A. Maillard and S. Mannor. Latent bandits. In *International Conference on Machine Learning*, pages 136–144, 2014.

[32] R. S. Mamon and R. J. Elliott. *Hidden Markov models in finance*, volume 4. Springer, 2007.

[33] A. J. Mersereau, P. Rusmevichientong, and J. N. Tsitsiklis. A structured multiarmed bandit problem and the greedy policy. *IEEE Transactions on Automatic Control*, 54(12):2787–2802, 2009.

[34] R. Ortner and D. Ryabko. Online regret bounds for undiscounted continuous reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1763–1771, 2012.

[35] R. Ortner, D. Ryabko, P. Auer, and R. Munos. Regret bounds for restless markov bandits. *Theoretical Computer Science*, 558:62–76, 2014.

[36] J. Qian, R. Fruit, M. Pirotta, and A. Lazaric. Exploration bonus for regret minimization in undiscounted discrete and continuous markov decision processes. *arXiv preprint arXiv:1812.04363*, 2018.

[37] S. M. Ross. Arbitrary state markovian decision processes. *The Annals of Mathematical Statistics*, 39(6):2118–2122, 1968.

[38] N. Saldi, S. Yüksel, and T. Linder. On the asymptotic optimality of finite approximations to markov decision processes with borel spaces. *Mathematics of Operations Research*, 42(4):945–978, 2017.

[39] H. Sharma, M. Jafarnia-Jahromi, and R. Jain. Approximate relative value learning for average-reward continuous state mdps. In *Uncertainty in Artificial Intelligence*, pages 956–964. PMLR, 2020.

[40] A. Slivkins. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*, 2019.

[41] A. Slivkins and E. Upfal. Adapting to a changing environment: the brownian restless bandits. In *Conference on Learning Theory*, pages 343–354, 2008.

[42] H. Yu and D. P. Bertsekas. Discretized approximations for pomdp with average cost. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 619–627. AUAI Press, 2004.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2.1
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 5
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]