IEEE Open Journal of Signal Processing

Received 9 January 2024; revised 30 March 2024; accepted 2 April 2024. Date of publication 16 April 2024;
date of current version 25 June 2024. The review of this article was arranged by Associate Editor R. Thilak Rajan.

Digital Object Identifier 10.1109/OJSP.2024.3389809

Contextual Multi-Armed Bandit With Costly Feature Observation in Non-Stationary Environments

SAEED GHOORCHIAN [1], EVGENII KORTUKOV [2], AND SETAREH MAGHSUDI [1]

[1]Faculty of Electrical Engineering and Information Technology, Ruhr-University Bochum, 44801 Bochum, Germany
[2]Faculty of Mathematics and Natural Sciences, Tübingen University, 72074 Tübingen, Germany

CORRESPONDING AUTHOR: SAEED GHOORCHIAN (email: saeed.ghoorchian@uni-tuebingen.de).

This work was supported by the German Federal Ministry of Education and Research (BMBF) under Grant 01IS20051.
This article has supplementary downloadable material available at https://doi.org/10.1109/OJSP.2024.3389809, provided by the authors.

ABSTRACT Maximizing long-term rewards is the primary goal in sequential decision-making problems. The majority of existing methods assume that side information is freely available, enabling the learning agent to observe all features' states before making a decision. In real-world problems, however, collecting beneficial information is often costly. That implies that, besides individual arms' reward, learning the observations of the features' states is essential to improve the decision-making strategy. The problem is aggravated in a non-stationary environment where reward and cost distributions undergo abrupt changes over time. To address the aforementioned dual learning problem, we extend the contextual bandit setting and allow the agent to observe subsets of features' states. The objective is to maximize the long-term average gain, which is the difference between the accumulated rewards and the paid costs on average. Therefore, the agent faces a trade-off between minimizing the cost of information acquisition and possibly improving the decision-making process using the obtained information. To this end, we develop an algorithm that guarantees a sublinear regret in time. Numerical results demonstrate the superiority of our proposed policy in a real-world scenario.

INDEX TERMS Contextual multi-armed bandit, non-stationary process, online learning, costly information acquisition.

I. INTRODUCTION

In a sequential decision-making problem, an agent takes action over consecutive rounds of play to optimize a long-term metric. Over the past decades, a large body of literature develop decision-making policies that deal with such optimization problems under various constraints [1], [2]. In most cases, particularly in the era of Big Data, the proposed methods postulate the possibility of information acquisition with no limit and for free. In reality, however, access to side information is challenging; collecting information might be costly. For example, in online advertising problems, the advertiser can purchase information about target users to display personalized ads. As another example, in medical contexts, obtaining information for treatment recommendations mainly requires additional tests that are time- and money-consuming. Thus, it is essential to develop algorithms that can learn the optimal observations and actions simultaneously.

Real-world problems frequently appear in non-stationary environments. For instance, in the application of personalized news recommendation, user preferences over news can change over time and exhibit various seasonality patterns [3]. As another example, in the wireless network routing problem, the quality and availability of each link may change over time due to network congestion or maintenance [4]. The non-stationarity in the spread pattern of pandemics such as COVID-19 is also an example, as the average number of infected individuals changes over time due to a given region's geographical- and demographical characteristics [5]. The dual learning problem described above becomes significantly more

challenging when the environment changes. In fact, in a non-stationary environment, the value of obtained information, such as received action's feedback or paid observation's cost, before a change in the environment might become obsolete after the change occurs. Therefore, the agent has to constantly adapt her strategy and improve the decision-making process to comply faster with the changes in the environment, while she simultaneously performs the aforementioned dual learning task.

We address the mentioned challenges by using the Multi-Armed Bandit (MAB) [6] framework, where a learning agent selects an arm at sequential decision-making rounds and the environment reveals a feedback drawn from some unknown probability distribution. In this setting, the agent experiences the exploration-exploitation dilemma, where the decision has to be made between exploring options to acquire new knowledge and selecting an option by exploiting the existing knowledge [7]. In a contextual MAB problem, the agent has additional access to some side information and is able to observe this contextual information before making decision at each round. However, in practice, such contextual information is not always readily available to the agent, but rather it has to be acquired in exchange for a cost.

In this paper, we model the described problem using the contextual bandit setting and introduce the non-stationary costly contextual bandit problem, which we call it NCC problem for short. We propose and analyze an algorithm to solve the NCC problem. Our proposed algorithm can be considered as a variant of the UCRL2 algorithm [8]. Moreover, it uses a sliding window to estimate the non-stationary rewards and costs in abruptly changing environments. We prove that our algorithm achieves a sublinear regret bound in time. We validate our solution on a real-world problem of ranking nursery school applications. The results demonstrate the superiority of our algorithm compared to several benchmarks. Our NCC model and solution effectively address the challenges in various real-world applications. Random costs in our setting correspond to changing real-world conditions when acquiring information for decision-making. For example, in a web-based recommender system, the prices of services that provide data, and the availability of compute and network infrastructure, are unstable. Hence, the cost of obtaining information varies over time. As another example, in a mobile edge computing problem, random costs result from the changing network conditions and job arrival rates at edge servers. In addition, non-stationarity, which is allowed by our framework, appears frequently in real-world scenarios. To list a few examples, it can correspond to alterations in trends and user preferences for a recommender system, distributional shifts in stock trading, or changing the average number of individuals exposed to COVID-19 over time due to a given region's geographical- and demographical characteristics. Evidently, in any real-world scenario, the ability to adapt to changes in both rewards and costs increases the system's performance.

In summary, the contributions of this paper are as follows.

- We formulate the non-stationary costly contextual (NCC) bandit problem where state observations are costly, reward and cost functions can take any (linear or nonlinear) form, and their corresponding generating processes are piece-wise stationary.
- We propose the NCC-UCRL2 algorithm for learning the state observations and actions simultaneously. Our proposed algorithm is applicable to solve many real-world problems, such as online advertising and stock trading.
- Theoretically, we analyze the regret performance of NCC-UCRL2 in stationary and non-stationary environments. We prove that NCC-UCRL2 achieves a sublinear regret in time.
- We demonstrate the superior performance of our proposed algorithm through numerical experiments and compare it with several benchmark algorithms.

A. RELATED WORKS

Non-stationary multi-armed bandits have attracted intensive attention in the past years, both from the theory [9], [10], [11], [12], [13] and the application [14], [15], [16], [17] side. Potential application domains span across different fields, including online recommender systems [3], [15], [16], [17], hyperparameter optimization [18], virtual reality for rehabilitation [19], split liver transplantation allocation [20], evaluation of information retrieval systems [21], or targeted Covid-19 border testing of travelers [22]. The state-of-the-art methods in non-stationary bandits either do not consider access to contextual information or do not assume costly information acquisition. In the seminal work of [9], the authors use a sliding window or a discount factor to estimate the rewards with piece-wise stationary generating processes. However, they only consider non-stationarity confined to a finite-number of change-points. Reference [23] extends this framework by considering evolution of mean rewards constrained by a variation budget. The authors also derive the connection between the amount of variation and minimal regret achievable in such a setting. [13] studies the linear stochastic bandit in a drifting environment while also considering a variation budget. The authors propose an Upper Confidence Bound (UCB)-based algorithm that adapts to reward changes using a sliding window and a Bandit-over-Bandit framework for tuning the proposed algorithm's parameter adaptively. The authors in [12] study linear stochastic bandits in abruptly changing and slowly varying environments. They utilize exponentially increasing weights of observations to reduce the influence of past observations with time, thereby adapting to environmental changes. In [17], the authors consider a contextual bandit problem and use two sliding windows to detect changes in reward distributions. If the rewards inside the second window are not predictable with high accuracy from observations inside the first window, the proposed algorithm considers a new change point. The observations since the last change point are used to select arms. Besides, [15] uses Gaussian random walks to model the non-stationarity in underlying reward-generating

processes. Online inference based on particle learning is applied to fit the bandit parameters sequentially. Moreover, [16] proposes a hierarchical bandit algorithm, which maintains a suite of bandit models that estimate the reward distributions using a subset of observations. A higher level bandit model measures if the prediction error of lower level models exceeds some threshold, discards them accordingly, and creates new ones. Further, [10] and [11] study the general non-stationary contextual MAB problem and propose algorithms that achieve sublinear regret bounds without the knowledge of the number of change points. In addition, [24], [25] study an online learning problem where the unknown model parameters follow a Markov jump process. The authors investigate various optimization objectives based on cost minimization and revenue (profit) maximization, and propose an online learning policy for the considered objectives. In [26], the authors investigate a setting related to ours, namely the switching-MDP problem. In the formulated problem, a certain number of abrupt changes in transition probabilities and reward distributions can occur. They develop a sliding window-based algorithm based on the UCRL2 policy [8], and derive two sublinear regret bounds for known- and unknown number of changes.

However, none of the works mentioned above consider costly information acquisition. Our paper, in contrast, focuses on non-stationary costly contextual bandits in abruptly changing environments with general (linear or nonlinear) reward and cost functions. Our proposed algorithm achieves sublinear regret by adapting to reward and cost distribution drifts, conditioned on tuning the sliding window size.

Costly features in online learning problems have been addressed both in the full information setting [27], [28], [29], and in the bandit setting [30]. However, the existing methods with bandit feedback either do not model the cost as a random variable or do not take into account the non-stationarity of the environment. Reference [30] is the most relevant work to ours. The authors consider a stationary contextual bandit problem where observing features' states is costly. However, the costs have constant values and the reward-generating processes are stationary. In contrast, we assume that the costs are random variables and the environment is non-stationary where reward and cost distributions drift abruptly at some change points. Our approach shall not be mistaken for MAB problems with paid observations [31], where the agent can observe the rewards of any subset of arms after paying the costs at each round. In contrast, in our work, we allow for feature vectors and assume that observing feature's states is costly.

Another related area of research is budget-constrained learning, where feature selection is adaptive. For example, the authors in [32] consider linear regression models under local and global constraints on the number of observed features. They propose an algorithm that actively chooses the features to observe for each data sample. As another example, the authors in [33] consider linear regression with a budget on the number of feature observations for each data sample. They analyze the number of required samples for the model with partial information to attain the same error as that with complete information. Unlike our approach, these works consider a batch learning setting with the free observation of a limited number of features. Besides, in [28], the authors investigate an online classification problem with a per-sample budget for observing features, where features have various costs. They propose a deep reinforcement learning algorithm to solve the problem. [34] studies a contextual bandit problem in which the agent has a fixed budget on the number of features she can observe before choosing an arm. The authors take advantage of Thompson sampling and propose an algorithm that works in stationary and non-stationary environments. However, they do not provide regret analysis for the proposed method. Compared to the aforementioned works, we do not assume a budget constraint; nonetheless, the agent attempts to minimize the total cost of observing features' states. Therefore, in our proposed method, the agent adaptively selects the features and learns the optimal policy from limited information.

The rest of the paper is as follows. We formulate the NCC bandit problem in Section II. We describe our proposed method, NCC-UCRL2, in Section III. In Section IV, we analyze the performance of NCC-UCRL2 theoretically. Section V includes numerical evaluation, and Section VI concludes the paper.

II. PROBLEM FORMULATION

Let $\mathcal{A} = \{1, 2, \ldots, A\}$ denote the set of *actions*. $\mathcal{D} = \{1, 2, \ldots, D\}$ represents a finite set of *features*. Each feature $i \in \mathcal{D}$ has some random state $\boldsymbol{\Phi}[i] \in \mathcal{X}_i$, where $\mathcal{X}_i$ denotes a finite set of states for feature i. We collect the random features' states of all the features in the random state vector $\boldsymbol{\Phi} = [\boldsymbol{\Phi}[1], \boldsymbol{\Phi}[2], \ldots, \boldsymbol{\Phi}[D]] \in \mathcal{X} = \bigotimes_{i\in\mathcal{D}} \mathcal{X}_i$. Let $\boldsymbol{\phi}$ be a realization of the random state vector, which is drawn from a fixed but unknown distribution. $\mathbb{P}[\boldsymbol{\Phi} = \boldsymbol{\phi}]$ shows the probability of state vector $\boldsymbol{\phi}$ being realized.

At each time t, the environment draws a *state vector* $\boldsymbol{\phi}_t = [\boldsymbol{\phi}_t[1], \boldsymbol{\phi}_t[2], \ldots, \boldsymbol{\phi}_t[D]]$. The agent can select a subset of features $\mathcal{I}_t \subseteq \mathcal{D}$, called the *observation set*, for costly observation. Other elements of the state vector remain unknown. When $|\mathcal{I}_t| = 0$, i.e., $\mathcal{I}_t = \emptyset$, none of features' states are observed at time t. We use $\mathcal{P}(\mathcal{D})$ to represent the power set of $\mathcal{D}$ that includes all possible observation sets, i.e., $\mathcal{P}(\mathcal{D}) = \{\mathcal{I} \subseteq \mathcal{D} \,|\, 0 \leq |\mathcal{I}| \leq D\}$. Besides, the *partial state vector* $\boldsymbol{\psi}_t = [\boldsymbol{\psi}_t[1], \boldsymbol{\psi}_t[2], \ldots, \boldsymbol{\psi}_t[D]]$ can be represented as

$$\boldsymbol{\psi}_t[i] = \begin{cases} \boldsymbol{\phi}_t[i], & \text{if } i \in \mathcal{I}_t, \\ ?, & \text{if } i \notin \mathcal{I}_t, \end{cases} \tag{1}$$

where ? indicates the corresponding feature's state is missing. Let $\mathcal{D}(\boldsymbol{\psi}) = \{i \in \mathcal{D} \,|\, \boldsymbol{\psi}[i] \neq ?\}$ represent the *domain set* of a partial state vector $\boldsymbol{\psi}$. By $\boldsymbol{\Psi}^+(\mathcal{I}) = \{\boldsymbol{\psi} \,|\, \mathcal{D}(\boldsymbol{\psi}) = \mathcal{I}\}$, we denote the set of all possible partial state vectors whose domain set is equal to the observation set $\mathcal{I}$. Therefore, $\boldsymbol{\Psi} = \bigcup_{\mathcal{I}\subseteq\mathcal{D}} \boldsymbol{\Psi}^+(\mathcal{I})$ denotes the set of all possible partial state vectors. Furthermore, we define a partial state vector $\boldsymbol{\psi}$ to be

consistent with $\boldsymbol{\phi}$ if $\boldsymbol{\psi}[i] = \boldsymbol{\phi}[i]$, $\forall i \in \mathcal{D}(\boldsymbol{\psi})$. We use $\boldsymbol{\phi} \simeq \boldsymbol{\psi}$ to show that $\boldsymbol{\psi}$ is consistent with $\boldsymbol{\phi}$. Moreover, $\boldsymbol{\psi}$ is a *substate* of $\boldsymbol{\psi}'$ if both the partial state vectors $\boldsymbol{\psi}$ and $\boldsymbol{\psi}'$ are consistent with $\boldsymbol{\phi}$ and $\mathcal{D}(\boldsymbol{\psi}) \subseteq \mathcal{D}(\boldsymbol{\psi}')$. We use $\boldsymbol{\psi} \preceq \boldsymbol{\psi}'$ to show that $\boldsymbol{\psi}$ is a substate of $\boldsymbol{\psi}'$. For every $i \in \mathcal{I}_t$, $\boldsymbol{c}_t[i] \in [0, 1]$ shows the random cost to observe $\boldsymbol{\phi}_t[i]$, which follows an unknown probability distribution with mean $\bar{\boldsymbol{c}}_t[i]$. Also, by $\boldsymbol{c}_t = [\boldsymbol{c}_t[1], \boldsymbol{c}_t[2], \ldots, \boldsymbol{c}_t[D]]$ and $\bar{\boldsymbol{c}}_t = [\bar{\boldsymbol{c}}_t[1], \bar{\boldsymbol{c}}_t[2], \ldots, \bar{\boldsymbol{c}}_t[D]]$, we denote the *cost vector* and the *mean cost vector* of all features at time t, respectively.

At each time t, the agent follows a policy π_t to select an observation set $\mathcal{I}_t$ and an action a_t. Therefore, we define the *policy* at time t using an ordered pair $\pi_t = (\mathcal{I}_t, h_t)$, where $h_t : \boldsymbol{\Psi}^+(\mathcal{I}_t) \to \mathcal{A}$ denotes an adaptive action selection strategy that maps a partial state vector $\boldsymbol{\psi}_t \in \boldsymbol{\Psi}^+(\mathcal{I}_t)$ to an action $a_t \in \mathcal{A}$. The agent then receives a random reward $r_t \in [0, 1]$ whose distribution is unknown a priori. We define the unknown expected reward function as $\bar{r}_t : \mathcal{A} \times \mathcal{X} \to [0, 1]$; hence $\bar{r}_t(a_t, \boldsymbol{\phi}_t)$ is the expected reward of action a_t at time t when the state vector is $\boldsymbol{\phi}_t$. The generating processes of rewards and costs are piece-wise stationary so that there exist Υ_T time instants before a time horizon T where at least one of the mean rewards or mean costs changes abruptly. We define the marginal probabilities and expected rewards of partial state vectors using the definition of probability distribution and expected reward for the state vectors. The marginal probability of the partial state vector $\boldsymbol{\psi}_t$ being realized at time t is defined as $p(\boldsymbol{\psi}_t) = \mathbb{P}[\boldsymbol{\Phi}_t \simeq \boldsymbol{\psi}_t]$. Moreover, $\bar{r}_t(a_t, \boldsymbol{\psi}_t) = \mathbb{E}[\bar{r}_t(a_t, \boldsymbol{\Phi}_t) \,|\, \boldsymbol{\Phi}_t \simeq \boldsymbol{\psi}_t]$ indicates the marginal expected reward of action a_t when the partial state vector $\boldsymbol{\psi}_t$ is observed. Therefore, for a fixed observation set $\mathcal{I}$, it holds that $\sum_{\boldsymbol{\psi} \in \boldsymbol{\Psi}^+(\mathcal{I})} p(\boldsymbol{\psi}) = 1$. At each time t, the sequence of the events in NCC bandit problem is summarized in **Game Protocol 1**.

Definition 1: A piece-wise stationary random process is a random process whose instantaneous outcomes are drawn from some probability distribution that remains time-invariant over disjoint time intervals $[t_i, t_{i+1})$, $i = 1, 2, \ldots, \Upsilon_T$, but changes from one interval to the other [35], [36]. In other words, a piece-wise stationary random process exhibits different stationary characteristics over distinct intervals. As mentioned before, in our proposed NCC problem, the mean rewards and mean costs are piece-wise constant concerning the time t; they remain constant unless they experience a change at some specific time(s), referred to as *change point(s)*. Naturally, the change points are not necessarily identical, i.e., the mean rewards and mean costs do not always change simultaneously. Therefore, by the definition above, the random processes of rewards and costs are piece-wise stationary.

The *expected gain* of the agent following the policy $\pi = (\mathcal{I}, h)$ at time t yields

$$\rho_t^{\pi} = \sum_{\boldsymbol{\psi} \in \boldsymbol{\Psi}^+(\mathcal{I})} p(\boldsymbol{\psi}) \bar{r}_t(h(\boldsymbol{\psi}), \boldsymbol{\psi}) - \sum_{i \in \mathcal{I}} \bar{\boldsymbol{c}}_t[i]. \tag{2}$$

Game Protocol 1 Sequence of Events in the NCC Bandit Problem.

Step 1: The environment reveals a state vector $\boldsymbol{\phi}_t$ according to a fixed but unknown probability distribution $p(\cdot)$. The agent does not know $\boldsymbol{\phi}_t$ initially.

Step 2: The agent selects an observation set $\mathcal{I}_t \subseteq \mathcal{D}$ to observe their states. The partial state vector $\boldsymbol{\psi}_t$ corresponding to the observation set $\mathcal{I}_t$ is revealed, while other features' states remain unknown.

Step 2: For every $i \in \mathcal{I}_t$, the agent pays a random cost $\boldsymbol{c}_t[i] \in [0, 1]$ which follows an unknown probability distribution with mean $\bar{\boldsymbol{c}}_t[i]$. The generating processes of costs are piece-wise stationary.

Step 3: Based on the observed partial state vector $\boldsymbol{\psi}_t$, the agent selects an action $a_t \in \mathcal{A}$. Then, the agent receives a random reward $r_t \in [0, 1]$ whose distribution is unknown with the unknown expected reward function $\bar{r}_t : \mathcal{A} \times \mathcal{X} \to [0, 1]$. Similar to costs, the generating processes of rewards are piece-wise stationary.

TABLE 1 Summary of Notations

Notation	Definition
$\mathcal{A}$	Set of actions
$\mathcal{D}$	Set of features
$\boldsymbol{\phi}_t$	Unknown state vector at time t
$\mathcal{I}_t$	Observation set of selected features at time t
$\boldsymbol{\psi}_t$	Partial state vector observed by the agent at time t
a_t	Action of the agent at time t
r_t	Reward at time t
$\boldsymbol{c}_t[i]$	Cost of state observation for feature $i \in \mathcal{D}$ at time t
ρ_t^{π}	Expected gain of policy π
$\mathcal{D}(\boldsymbol{\psi})$	Domain set of partial state vector $\boldsymbol{\psi}$
$\boldsymbol{\Psi}^+(\mathcal{I})$	Set of all partial state vectors with domain $\mathcal{I}$
$\boldsymbol{\Psi}$	Set of all partial state vectors

In words, the expected gain of the agent that follows a policy π at time t is the expected reward of π received by the agent at time t minus the expected cost of π incurred by the agent due to state observation at time t. Let Π denote the set of all feasible policies defined as

$$\Pi = \{(\mathcal{I}, h) | \mathcal{I} \in \mathcal{P}(\mathcal{D})\}. \tag{3}$$

Therefore, the optimal policy $\pi_t^* = (\mathcal{I}_t^*, h_t^*)$ at time t is given by

$$\pi_t^* = \arg\max_{\pi \in \Pi} \ \rho_t^{\pi}. \tag{4}$$

Moreover, the expected gain of the optimal policy at time t is denoted by $\rho_t^* = \rho_t^{\pi_t^*}$. We summarize the most important notations in **Table 1**.

The optimal policy (4) for NCC problem differs from the conventional optimal policies in the contextual bandit problems. Let $a_t^*(\boldsymbol{\psi}) = \arg\max_{a \in \mathcal{A}} \bar{r}_t(a, \boldsymbol{\psi})$ denote the best action for a given partial state vector $\boldsymbol{\psi}$. Moreover, define $\bar{r}_t^*(\boldsymbol{\psi}) =$

$\bar{r}_t(a_t^*(\boldsymbol{\psi}), \boldsymbol{\psi})$ as the expected reward of the best action when the partial state vector is $\boldsymbol{\psi}$. Moreover, for a fixed observation set $\mathcal{I}$, define a policy $\pi_t(\mathcal{I}) = (\mathcal{I}, a_t^*(\boldsymbol{\psi}))$ that selects the observation set $\mathcal{I}$ and the best action $a_t^*(\boldsymbol{\psi})$ for any $\boldsymbol{\psi} \in \boldsymbol{\Psi}^+(\mathcal{I})$ at time t. The expected gain of the policy $\pi_t(\mathcal{I})$ can be calculated as $V_t(\mathcal{I}) = \sum_{\boldsymbol{\psi} \in \boldsymbol{\Psi}^+(\mathcal{I})} p(\boldsymbol{\psi}) \bar{r}_t^*(\boldsymbol{\psi}) - \sum_{i \in \mathcal{I}} \bar{c}_t[i]$. Then, the optimal policy $\pi_t^* = (\mathcal{I}_t^*, h_t^*)$ defined in (4) can be obtained by

$$\mathcal{I}_t^* = \arg\max_{\mathcal{I} \in \mathcal{P}(\mathcal{D})} V_t(\mathcal{I}),$$

$$h_t^*(\boldsymbol{\psi}) = \arg\max_{a \in \mathcal{A}} \bar{r}_t(a, \boldsymbol{\psi}). \tag{5}$$

We observe that $\rho_t^* = V_t(\mathcal{I}_t^*)$, which means the optimal policy (4) achieves the highest expected gain at each time t among all the policies $\pi_t(\mathcal{I})$.

Ideally, the agent aims at maximizing the total expected gain over the time horizon T. Alternatively, the agent's goal is to minimize the *expected regret* over the time horizon T, defined as the difference between the accumulated expected gain of the optimal policy and that of applied policy, i.e., the one that the agent follows. Formally, the expected regret is defined as

$$\mathcal{R}_T(\Pi) = \sum_{t=1}^{T} \left[\rho_t^* - \rho_t^{\pi_t}\right]. \tag{6}$$

In the next section, we propose a policy to minimize the expected regret (6).

III. DECISION-MAKING STRATEGY

In this section, we propose our decision-making strategy to solve the NCC problem described in Section II. Our policy, presented in **Algorithm 1**, takes three types of confidence regions into account, for rewards, costs, and probabilities of partial state vectors. Since the random generating processes of rewards and costs are non-stationary, we use a sliding window of size $w > 0$ to estimate their mean values. At each time t, we define

$$\mathcal{T}_t(a, \boldsymbol{\psi}; w) = \{t - w < \tau < t \,|\, a_\tau = a \quad \boldsymbol{\psi}_\tau = \boldsymbol{\psi}\}, \tag{7}$$

$$\mathcal{T}_t(i; w) = \{t - w < \tau < t \,|\, i \in \mathcal{I}_\tau\}. \tag{8}$$

For each $a \in \mathcal{A}$ and $\boldsymbol{\psi} \in \boldsymbol{\Psi}$, we calculate the empirical average of rewards at time t by

$$\hat{r}_t(a, \boldsymbol{\psi}) = \frac{1}{N_t(a, \boldsymbol{\psi}; w)} \sum_{\tau \in \mathcal{T}_t(a, \boldsymbol{\psi}; w)} r_\tau, \tag{9}$$

where $N_t(a, \boldsymbol{\psi}; w) = \max\{1, |\mathcal{T}_t(a, \boldsymbol{\psi}; w)|\}$. Moreover, at each time t, we calculate the empirical average of costs for each $i \in \mathcal{D}$ by

$$\hat{\boldsymbol{c}}_t[i] = \frac{1}{N_t(i; w)} \sum_{\tau \in \mathcal{T}_t(i; w)} \boldsymbol{c}_\tau[i], \tag{10}$$

where $N_t(i; w) = \max\{1, |\mathcal{T}_t(i; w)|\}$.

Our policy uses the collected data to estimate the probabilities of partial state vectors; that is, after observing the partial state vector $\boldsymbol{\psi}_t$, the agent uses it to update the estimate of the probability of $\boldsymbol{\psi}_t$ and the probabilities of all the substates of $\boldsymbol{\psi}_t$. However, the agent cannot use the obtained reward at time t to update the estimate of mean reward for action a_t and the sub-states of $\boldsymbol{\psi}_t$, since it introduces a bias into the mean reward estimation. Therefore, we define

Algorithm 1: NCC-UCRL2.

Input: Window size w.

1: **Initialize:** $\forall a \in \mathcal{A}, \forall \boldsymbol{\psi} \in \boldsymbol{\Psi}, \forall i \in \mathcal{D}, \forall \mathcal{I} \in \mathcal{P}(\mathcal{D})$: $\mathcal{T}_1(a, \boldsymbol{\psi}; w) = \emptyset$, $\mathcal{T}_1(i; w) = \emptyset$, $\mathcal{T}_1(\mathcal{I}) = \emptyset$, $\mathcal{T}_1(\mathcal{I}, \boldsymbol{\psi}) = \emptyset$.
2: **for** $t = 1, \ldots, T$ **do**
3: Compute $\hat{r}_t(a, \boldsymbol{\psi})$, $\forall a \in \mathcal{A}, \forall \boldsymbol{\psi} \in \boldsymbol{\Psi}$, using (9).
4: Compute $\hat{\boldsymbol{c}}_t[i]$, $\forall i \in \mathcal{D}$, using (10).
5: Compute $\hat{p}_t(\boldsymbol{\psi})$, $\forall \boldsymbol{\psi} \in \boldsymbol{\Psi}$. using (13).
6: Solve Problem (17), $\forall \mathcal{I} \in \mathcal{P}(\mathcal{D})$, and obtain $\hat{V}_t(\mathcal{I})$.
7: Select the observation set $\hat{\mathcal{I}}_t$ that solves (18) and pay the cost $\sum_{i \in \hat{\mathcal{I}}_t} \boldsymbol{c}_t[i]$.
8: Determine the action selection strategy $\hat{h}_t(\boldsymbol{\psi})$ based on (19).
9: Observe the partial state vector $\boldsymbol{\psi}_t \in \boldsymbol{\Psi}^+(\hat{\mathcal{I}}_t)$.
10: Select the action $a_t = \hat{h}_t(\boldsymbol{\psi}_t)$ and observe the reward r_t.
11: Update $\mathcal{T}_t(\mathcal{D}(\boldsymbol{\psi}))$ and $\mathcal{T}_t(\mathcal{D}(\boldsymbol{\psi}), \boldsymbol{\psi})$, $\forall \boldsymbol{\psi}$ s.t. $\boldsymbol{\psi} \preceq \boldsymbol{\psi}_t$.
12: Update $\mathcal{T}_t(a_t, \boldsymbol{\psi}_t; w)$.
13: Update $\mathcal{T}_t(i; w)$, $\forall i \in \hat{\mathcal{I}}_t$.
14: **end for**

$$\mathcal{T}_t(\mathcal{I}) = \{\tau < t \,|\, \mathcal{I} \subseteq \mathcal{I}_\tau\}, \tag{11}$$

$$\mathcal{T}_t(\mathcal{I}, \boldsymbol{\psi}) = \begin{cases} \{\tau < t \,|\, \mathcal{I} \subseteq \mathcal{I}_\tau \quad \boldsymbol{\psi} \preceq \boldsymbol{\psi}_\tau\}, & \boldsymbol{\psi} \in \boldsymbol{\Psi}^+(\mathcal{I}), \\ \emptyset, & \boldsymbol{\psi} \notin \boldsymbol{\Psi}^+(\mathcal{I}). \end{cases} \tag{12}$$

Then, we estimate the probability for each partial state vector $\boldsymbol{\psi} \in \boldsymbol{\Psi}$ at time t as

$$\hat{p}_t(\boldsymbol{\psi}) = \frac{N_t(\mathcal{D}(\boldsymbol{\psi}), \boldsymbol{\psi})}{N_t(\mathcal{D}(\boldsymbol{\psi}))}, \tag{13}$$

where $N_t(\mathcal{I}, \boldsymbol{\psi}) = \max\{1, |\mathcal{T}_t(\mathcal{I}, \boldsymbol{\psi})|\}$ and $N_t(\mathcal{I}) = \max\{1, |\mathcal{T}_t(\mathcal{I})|\}$.

When searching for the optimal observation set and action, we add high-probability confidence bounds to the aforementioned estimates. Let $\boldsymbol{\Psi}_{tot} = \sum_{\mathcal{I} \in \mathcal{P}(\mathcal{D})} |\boldsymbol{\Psi}^+(\mathcal{I})|$ and $\delta > 0$. For each action $a \in \mathcal{A}$ and partial state vector $\boldsymbol{\psi} \in \boldsymbol{\Psi}$, we define

$$\tilde{r}_t(a, \boldsymbol{\psi}) = \hat{r}_t(a, \boldsymbol{\psi}) + C_t(a, \boldsymbol{\psi}; w), \tag{14}$$

where $C_t(a, \boldsymbol{\psi}; w) = \min\left\{1, \sqrt{\frac{\log(TA\boldsymbol{\Psi}_{tot} w/\delta)}{N_t(a, \boldsymbol{\psi}; w)}}\right\}$. Moreover, for each feature $i \in \mathcal{D}$, we define

$$\tilde{\boldsymbol{c}}_t[i] = \hat{\boldsymbol{c}}_t[i] - C_t(i; w), \tag{15}$$

where $C_t(i; w) = \min\left\{1, \sqrt{\frac{2\log(TDw/\delta)}{N_t(i; w)}}\right\}$. The optimistic gain at time t can be found by searching for partial state vector

probabilities over a high-probability space and a policy that solves

$$\underset{\substack{\pi=(\mathcal{I},h),\\ q\in\Delta_{|\boldsymbol{\Psi}^+(\mathcal{I})|}}}{\text{maximize}} \left\{ \sum_{\boldsymbol{\psi}\in\boldsymbol{\Psi}^+(\mathcal{I})} q(\boldsymbol{\psi})\tilde{r}_t(h(\boldsymbol{\psi}),\boldsymbol{\psi}) - \sum_{i\in\mathcal{I}} \tilde{\boldsymbol{c}}_t[i] \;\middle|\; \sum_{\boldsymbol{\psi}\in\boldsymbol{\Psi}^+(\mathcal{I})} |q(\boldsymbol{\psi}) - \hat{p}_t(\boldsymbol{\psi})| \le C_t(\mathcal{I}) \right\}, \quad (16)$$

where $C_t(\mathcal{I}) = \min\left\{1, \sqrt{\frac{2\boldsymbol{\Psi}_{tot}\log(2T|\mathcal{P}(\mathcal{D})|/\delta)}{N_t(\mathcal{I})}}\right\}$ and $\Delta_{|\boldsymbol{\Psi}^+(\mathcal{I})|}$ is a simplex in $|\boldsymbol{\Psi}^+(\mathcal{I})|$ dimensions. The optimization problem (16) can be reduced to the following optimization problem (See Section I of supplementary material for details).

$$\hat{V}_t(\mathcal{I}) = \underset{q\in\Delta_{|\boldsymbol{\Psi}^+(\mathcal{I})|}}{\text{maximize}} \left\{ \sum_{\boldsymbol{\psi}\in\boldsymbol{\Psi}^+(\mathcal{I})} q(\boldsymbol{\psi})\tilde{r}^*_t(\boldsymbol{\psi}) - \sum_{i\in\mathcal{I}} \tilde{\boldsymbol{c}}_t[i] \;\middle|\; \sum_{\boldsymbol{\psi}\in\boldsymbol{\Psi}^+(\mathcal{I})} |q(\boldsymbol{\psi}) - \hat{p}_t(\boldsymbol{\psi})| \le C_t(\mathcal{I}) \right\}, \quad (17)$$

where $\tilde{r}^*_t(\boldsymbol{\psi}) = \max_{a\in\mathcal{A}} \tilde{r}_t(a,\boldsymbol{\psi})$ is the optimistic reward estimate of the partial state vector $\boldsymbol{\psi}$ at time t. Problem (17) is solved by ranging the value of q over the plausible candidate set of probabilities for $p(\boldsymbol{\psi})$. We denote the value of q that solves (17) at time t by $\tilde{p}_t(\boldsymbol{\psi})$. Note that, for each $\mathcal{I}$, the probability $\tilde{p}_t(\boldsymbol{\psi})$ denotes the optimistic probability estimate of the partial state vector $\boldsymbol{\psi}\in\boldsymbol{\Psi}^+(\mathcal{I})$ at time t. Moreover, $\hat{V}_t(\mathcal{I})$ represents the optimistic gain of a policy $\pi_t(\mathcal{I}) = (\mathcal{I}, \hat{h}_t(\boldsymbol{\psi}))$ that selects the observation set $\mathcal{I}$ and the action $\hat{h}_t(\boldsymbol{\psi})$ for any $\boldsymbol{\psi}\in\boldsymbol{\Psi}^+(\mathcal{I})$ at time t.

At each time t, our algorithm solves (17) and acts optimistically by choosing the observation set and determining the action selection strategy as

$$\hat{\mathcal{I}}_t = \underset{\mathcal{I}\in\mathcal{P}(\mathcal{D})}{\arg\max}\ \hat{V}_t(\mathcal{I}), \quad (18)$$

and

$$\hat{h}_t(\boldsymbol{\psi}) = \underset{a\in\mathcal{A}}{\arg\max}\ \hat{r}_t(a,\boldsymbol{\psi}) + C_t(a,\boldsymbol{\psi};w), \quad (19)$$

respectively. Afterward, NCC-UCRL2 pays the costs corresponding to the selected observation set $\hat{\mathcal{I}}_t$, observes the partial state vector $\boldsymbol{\psi}_t\in\boldsymbol{\Psi}^+(\hat{\mathcal{I}}_t)$, and takes the action $a_t = \hat{h}_t(\boldsymbol{\psi}_t)$. Finally, it receives the corresponding reward r_t and updates the counters. The computational complexity of NCC-UCRL2 algorithm is $O(A\boldsymbol{\Psi}_{tot}T)$.

IV. THEORETICAL ANALYSIS

In this section, we analyze the regret performance of NCC-UCRL2 algorithm in stationary and non-stationary environments. We first prove an upper bound on the expected regret of our algorithm by assuming that the environment is stationary, i.e., there is no change point in the environment. In the stationary case, it is natural to choose $w = \Theta(T)$ to exploit the entire collected data for estimation of the mean rewards and mean costs. In this case, as expected, NCC-UCRL2 achieves a sublinear regret with respect to time.

Theorem 1: When the environment is stationary, i.e., $\Upsilon_T = 0$, with probability at least $1-3\delta$, the expected regret of NCC-UCRL2 is upper bounded as

$$\begin{aligned}\mathcal{R}_T(\Pi) &\le O\Bigg(T\left(\sqrt{\frac{A\boldsymbol{\Psi}_{tot}\log(TA\boldsymbol{\Psi}_{tot}w/\delta)}{w}} + D\sqrt{\frac{\log(TDw/\delta)}{w}}\right)\\ &+ \sqrt{T\log(1/\delta)}\Bigg(\sqrt{A\boldsymbol{\Psi}_{tot}\log(TA\boldsymbol{\Psi}_{tot}w/\delta)}\\ &+ D\sqrt{\log(TDw/\delta)}\Bigg)\\ &+ \sqrt{T|\mathcal{P}(D)|\boldsymbol{\Psi}_{tot}\log(T|\mathcal{P}(\mathcal{D})|/\delta)}\Bigg). \quad (20)\end{aligned}$$

In this case, choosing $w = T$ results in the following bound.

$$\begin{aligned}\mathcal{R}_T(\Pi) &\le O\Bigg(\left(1+\sqrt{\log(1/\delta)}\right)\Bigg(\sqrt{TA\boldsymbol{\Psi}_{tot}\log(TA\boldsymbol{\Psi}_{tot}/\delta)}\\ &+ D\sqrt{T\log(TD/\delta)}\Bigg)\\ &+ \sqrt{T|\mathcal{P}(D)|\boldsymbol{\Psi}_{tot}\log(T|\mathcal{P}(\mathcal{D})|/\delta)}\Bigg). \quad (21)\end{aligned}$$

Proof: See Section IV-A of supplementary material. ■

The proof of Theorem 1 is, to some extent, based on state-of-the-art techniques used in the literature to analyze regret bounds for optimistic bandit algorithms; nevertheless, some non-conventional parts appear in our derivation because we estimate the partial state probabilities using all observations, while the mean rewards and mean costs using the most recent ones in the window. Note that, in the optimization problem (16), we use *optimistic estimations* for rewards and partial state probabilities, whereas we rely on *pessimistic ones* for costs by using the lower confidence bound on the mean costs in (15). That results in several technical challenges in the theoretical analysis, for example, in *Lemma 3*, where we bound the probability of failure (See Section IV of supplementary material). Moreover, proving the bound in (50) is challenging as the algorithm can choose more than one feature at a time. Hence, in (50), we consider the worst case of observing all the D features' states at each time t.

In the next theorem, we establish an upper bound on the expected regret of NCC-UCRL2 in non-stationary environments. With the right choice of the window size, NCC-UCRL2 achieves sublinear regret in time. The regret analysis

TABLE 2 Comparison With Related Works

	NCC-UCRL2 (our work)	Sim-OOS [30]	PS-LinUCB [17]	SW-UCRL [26]
Costly Features	Yes	Yes	No	No
Non-stationary	Yes	No	Yes	Yes
Regret w.r.t Time	sublinear	sublinear	sublinear	sublinear

for the non-stationary case is based on the theoretical analysis in Theorem 1.

Theorem 2: When the environment is non-stationary, i.e., $\Upsilon_T > 0$, with probability at least $1 - 3\delta$, the expected regret of NCC-UCRL2 is upper bounded as

$$\begin{aligned}\mathcal{R}_T(\Pi) &\leq O\Bigg(w\Upsilon_T + T\Bigg(\sqrt{\frac{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)}{w}} \\ &\quad + D\sqrt{\frac{\log(TDw/\delta)}{w}}\Bigg) \\ &\quad + \sqrt{\Upsilon_T T\log(1/\delta)}\Bigg(\sqrt{A\Psi_{tot}\log(TA\Psi_{tot}w/\delta)} \\ &\quad + D\sqrt{\log(TDw/\delta)}\Bigg) \\ &\quad + \sqrt{T|\mathcal{P}(D)|\Psi_{tot}\log(T|\mathcal{P}(\mathcal{D})|/\delta)}\Bigg). \end{aligned} \tag{22}$$

In this case, choosing $w = (T/\Upsilon_T)^{2/3}$ results in the following bound.

$$\begin{aligned}\mathcal{R}_T(\Pi) \leq O\Bigg(&\left(T^{2/3}\Upsilon_T^{1/3} + \sqrt{\Upsilon_T T\log(1/\delta)}\right) \\ &\left(\sqrt{A\Psi_{tot}\log(TA\Psi_{tot}/\delta)} + D\sqrt{\log(TD/\delta)}\right) \\ &+ \sqrt{T|\mathcal{P}(D)|\Psi_{tot}\log(T|\mathcal{P}(\mathcal{D})|/\delta)}\Bigg). \end{aligned} \tag{23}$$

Proof: See Section IV-B of supplementary material.■

The analysis in Theorem 2 is based on Theorem 1. During the stationary phases, the algorithm suffers the same sublinear regret proved in Theorem 1. When experiencing a change point, the algorithm suffers an extra $O(w)$ regret, while the second term in (20) scales by a factor of $\sqrt{\Upsilon_T}$. Our algorithm does not require the knowledge of Υ_T and guarantees a sublinear regret bound with a proper choice of w, as given by (23).

Table 2 summarizes the comparison of our paper with the closest works.

Remark 1: In this paper, we study and analyze the general NCC bandit problem where the reward and cost might follow any linear or nonlinear function. Due to its generality, solving this problem comes at the price of potentially excessive computational burden: The complexity and regret bound of our proposed algorithm depend on the number of possible combinations of the features' states, which is sometimes large; Nonetheless, there are several approaches to mitigate the effect of such a term on the regret bound and computational complexity. Below, we elaborate on such methods:

- The complexity diminishes if the number of features or the number of features' states is small. That happens in numerous real-world problems where the features can be filtered out based on prior knowledge and/or the states can be quantized efficiently. For example, in a medical setting, the clinician limits the potentially useful tests to a specific small set. Moreover, the outcome of each test can be interpreted as healthy or not. As another example, in a wireless communication network, one can describe the channel state as high-quality or low-quality based on the QoS requirement.
- The complexity decreases if we limit the number of observations allowed at each time by considering a predetermined value for the maximum number of feature selections at each time.
- One can reduce the complexity by combining approximate sampling methods with feature selection strategies, for example, by defining Shapley values for features, estimating these values using Monte Carlo methods, and choosing features based on the approximated Shapley values.
- It is also beneficial to allow for restrictive assumptions on the space of reward and cost functions. In such cases, the dependence of the regret bound and computational complexity on the number of features and partial states diminishes. For example, the complexity decreases by considering linear reward functions, where the expected reward of each arm is a linear function of the contexts with some unknown coefficient [37].

V. NUMERICAL ANALYSIS

In this section, via numerical experiments, we provide more insights into the effects of costly features on the performance of learning algorithms. Besides, we clarify how our proposed algorithm mitigates the adverse effects by observing only a subset of features' states. Moreover, we show that our algorithm efficiently adapts to environmental changes. We also compare the performance of our algorithm with conventional benchmarks using a real-world dataset. The source code for our algorithm and experiments in this paper are publicly available.[1]

Benchmark Policies: We compare NCC-UCRL2 with the state-of-the-art contextual and context-agnostic algorithms. Contextual bandit algorithms in our experiment include **Sim-OOS** [30], **PS-LinUCB** [17], and **LinUCB** [37]. Sim-OOS is designed for bandit problems with fixed costs for features' states observation in stationary environments. PS-LinUCB is designed for piece-wise stationary environments, but it is

[1]Source code: https://github.com/saeedghoorchian/NCC-Bandits.git

cost-agnostic. LinUCB is the final contextual bandit algorithm that is neither designed for changing environments nor costly features. In our experiment, similar to our algorithm, Sim-OOS can select any subset of features for state observation at each time of play. As a result, at each time, they pay the corresponding cost only for those selected features. PS-LinUCB and LinUCB always observe all features' states. Hence, they pay the full cost vector. We consider **UCB1** and **ε-Greedy** [38] as context-agnostic benchmarks as standard methods despite their weakness due to being blind to contextual information. We also consider a **random** policy that selects an action uniformly at random at each time. Context-agnostic algorithms do not incur any costs and only collect the rewards.

Nursery Dataset: We assess the performance of our algorithm on the Nursery dataset from the UCI Machine Learning Repository [39]. The dataset, derived from a hierarchical decision support system, includes applications for nursery schools and their target ranks that prioritize the applications and determine whether the child is recommended to be admitted to a nursery school. The applications are described using features that represent the socioeconomic status of the family. We consider $D = 5$ features: i) Form of the family, ii) number of children, iii) financial standing of the family, iv) housing conditions, and v) health conditions of the applicant. In our experiment, we work with $A = 3$ target rank values ranging from 1 to 3 that indicate the given application is *not recommended*, *accepted with priority*, and *accepted with special priority*, respectively. Taking an action is equivalent to recommending one particular rank for the given application. The agent receives reward 1 if the correct rank is recommended, otherwise the reward is 0.

Experimental Setup: To simulate a piece-wise stationary reward generating process, we follow the approach proposed by [34]. At each change point, we shift all the target labels cyclically. This guarantees that the expected reward is piece-wise constant. In the context of decision support system for nursery school applications, such change points correspond to changes in preference of the decision-making authority over the applications.

We endow the features with random cost values. At each time t, the random cost of observation for each feature's state follows a normal distribution with a standard deviation of 0.001 and a piece-wise constant mean. We select the mean values of cost distributions uniformly at random from the interval [0.03; 0.08]. Therefore, the total observation cost of a full state vector at each time amounts to $15 - 40\&$; *percnt*; of the maximum reward. The range of costs are chosen based on two factors: i) It should be high enough to prevent the algorithm from observing all features' states at all times and, ii) low enough to incentivize the algorithm considerably to pay for state observation in order to find the optimal observations. In the nursery application ranking scenario, the state observation costs can be thought of as the efforts required to acquire the information about the applicant. Such efforts may include the time or other related expenses spent to obtain the information.

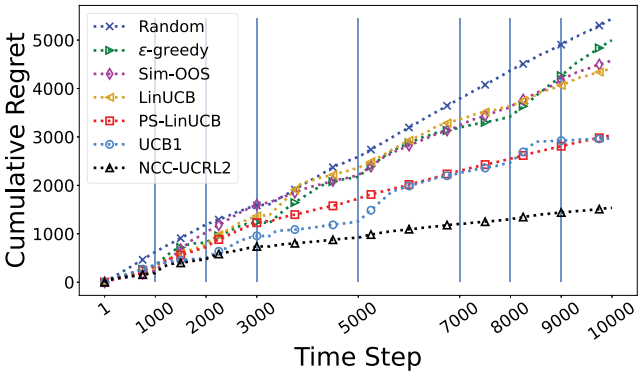

FIGURE 1. **Cumulative regret of different policies. Vertical lines show the change points.**

We split the data into train and validation (tuning) sets in approximately 80:20 ratio with 10000 and 2630 data samples, respectively. More specifically, we sample 2630 data points at random and use them to tune the parameters of algorithms. The parameters of those benchmark algorithms that are originally designed for stationary environments are tuned without introducing non-stationarity in the validation set. To tune the parameters of NCC-UCRL2 and PS-LinUCB, we consider 2 change points in mean rewards, but no change points in mean costs. For more details on the tuning process of the parameters, please see Section V of the supplementary material.

We run the experiment for $T = 10000$ time steps by revealing applications to the algorithms one at a time. We consider a maximum of $\Upsilon_T = 7$ change points in our experiment, with change points in the mean rewards and the mean costs at times $\{1000, 2000, 5000, 8000\}$ and $\{3000, 5000, 7000, 9000\}$, respectively. Note that the change points are not necessarily identical; the mean rewards and mean costs do not always change simultaneously at a change point. In Section V of the supplementary material, we elaborate more on the settings of mean rewards and mean costs. **Table 3** in the supplementary material lists the tuned parameters of algorithms used in our simulation. For NCC-UCRL2, we set $\delta = 0.04$ and choose the window parameter $w = 250$.

Remark 2: Assuming abrupt changes is standard in the literature concerning piece-wise stationary multi-armed bandits; nonetheless, it is essential to mention that in many real-world applications, the environment evolves gradually. The state-of-the-art algorithms, including ours, show an acceptable performance also in those scenarios; nevertheless, their efficiently degrades as detecting small changes is more troublesome and associated with frequent false alarms and missed detections. Besides, sometimes it is not vital to detect small changes because they degrade the performance only slightly. As such, the system would intentionally ignore small changes to maintain efficiency, leading to a model similar to the one considered here, i.e., abrupt changes.

Regret Comparison: We run the algorithms using the aforementioned setup. Fig. 1 depicts the trend of cumulative regret

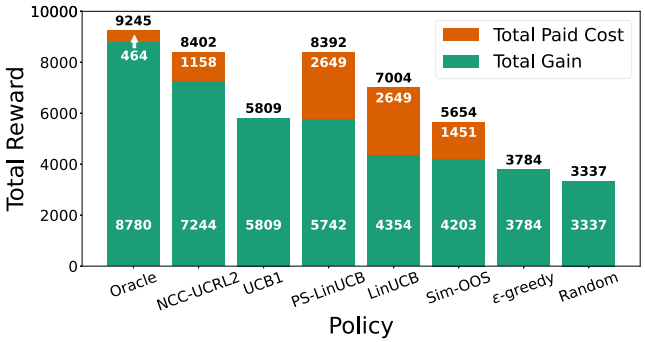

FIGURE 2. Total reward (number on top of bar), gain (number in green), and cost (number in brown) for each policy. Values are rounded to the nearest integers.

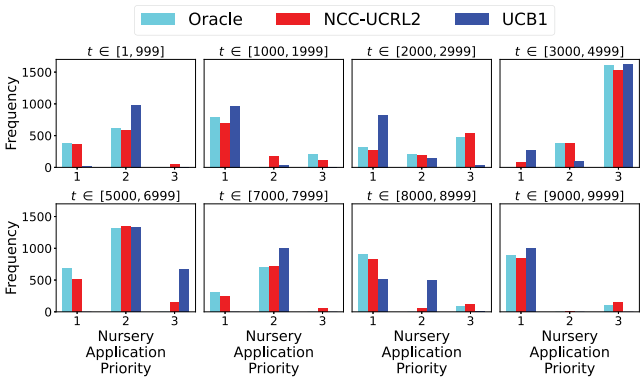

FIGURE 3. Comparison of priority recommendations of the optimal policy (oracle), NCC-UCRL2, and UCB1 in each stationary period.

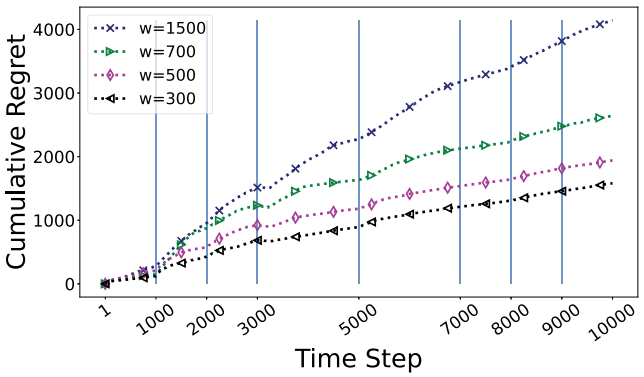

FIGURE 4. Cumulative regret of NCC-UCRL2 for different window parameters w.

over time for each policy. We average the results over 5 independent runs. Here, the instantaneous regret at each time is defined based on the instantaneous gain, which is the obtained reward minus the total paid observation costs at every round. As we see, NCC-UCRL2 detects the changes in the mean rewards or mean costs faster than all other policies and therefore has a superior performance. Besides, as NCC-UCRL2 uses only the last w observations to estimate the mean rewards and mean costs, it has a smooth curve around change points. These advantages are despite the fact that NCC-UCRL2 only observes a subset of features' states at each time.

Gain Comparison: In Fig. 2, we show the policies' total reward, gain, and cost. It also compares them with the optimal policy (oracle). In this figure, the height of each bar shows the total accumulated reward of each policy which is equal to the total gain (green part) plus the total cost (brown part). NCC-UCRL2 accumulates the highest rewards during the experiment among the benchmark policies. The accumulated reward of PS-LinUCB is almost the same as that of our algorithm; it receives only about 0.1% less reward than NCC-UCRL2. However, the total gain of PS-LinUCB is 20% lower due to higher paid costs as it observes all the features' states at all times. On the contrary, NCC-UCRL2 adaptively learns the optimal state observations while it observes only a fraction of features' states at each time. As a result, NCC-UCRL2 incurs less cost, hence a higher performance concerning the accumulated gain. The two counterparts of NCC-UCRL2 and PS-LinUCB that suit stationary environments, i.e., Sim-OOS and LinUCB, exhibit a similar pattern for the total costs; nevertheless, Sim-OOS achieves lower accumulated reward compared to LinUCB, which shows the importance of learning the optimal observations in a non-stationary environment. Note that Sim-OOS fails in our experiment as it does not consider the pessimistic selection of random costs and cannot adapt to drifts.

Adaptation to the Preference Volatility: In Fig. 3, we plot the histograms of nursery application priorities recommended by the optimal policy, NCC-UCRL2, and UCB1 for each of the stationary periods. Our algorithm closely follows the arm choice pattern of the optimal policy, which means that it can quickly adapt to changes in preference over applications. On the other hand, UCB1 cannot always adapt to sudden changes in the environment. We particularly consider UCB1 in this analysis to show the following: Although UCB1 achieves the second highest gain amongst the benchmarks, it fails to provide tailored recommendations when the environment parameters undergo abrupt changes.

Effect of Window Length w: Choosing the right window parameter w is crucial to ensure that the NCC-UCRL2 algorithm promptly adjusts the decision-making strategy after sudden changes while maintaining a good performance during stationary periods. The window size w can be chosen based on the change frequency. A smaller w allows for faster adaptation but reduces the performance during stationary periods due to exploiting fewer relevant data samples. In an environment with infrequent change points, a larger w is more suitable as it results in a better performance between change points, although the algorithm requires more storage space. Fig. 4 illustrates the trend of cumulative regret of our algorithm when running on the nursery dataset with different window parameters w. Based on our simulation's setting, we see that NCC-UCRL2 with smaller window sizes (around 300) results

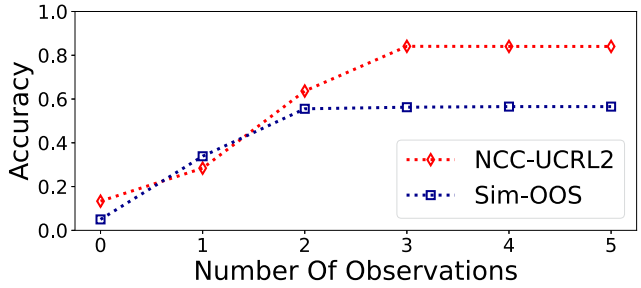

FIGURE 5. Accuracy for different number of observations.

in a much lower regret (e.g., compared to values more than 700).

Accuracy: To further analyze the performance of our algorithm, we define *accuracy* for the model based on the number of state observations. With ℓ observations, the accuracy yields $\left(\sum_{j=0}^{\ell}\sum_{t=1}^{T} r_t \mathbb{1}\{|\mathcal{I}_t| = j\}\right) / \left(\sum_{j=0}^{\ell}\sum_{t=1}^{T} \mathbb{1}\{|\mathcal{I}_t| = j\}\right)$. We use the term accuracy since, in our experiment, a reward of 1 implies the correct classification of a nursery application. [30] perform a similar analysis for Sim-OOS. Therefore, we plot the accuracy of NCC-UCRL2 and Sim-OOS for a different number of observations in Fig. 5, as these are the only algorithms that implement feature selection. For fewer observations, the accuracy of Sim-OOS is close to that of NCC-UCRL2, while NCC-UCRL2 achieves a higher accuracy as the number of observations increases. This again shows the importance of learning the optimal observations and demonstrates the superiority of our method.

VI. CONCLUSION

We introduced the NCC bandit framework, where information acquisition is costly and the environment is non-stationary. We developed a decision-making policy, namely NCC-UCRL2, that mitigates the effects of costs by observing only a subset of features. We proved that NCC-UCRL2 achieves a sublinear regret bound in time. Our proposed framework is applicable in several contexts, such as online advertising problems, medical treatment recommendations, edge computing, and stock trading. We applied our method to recommend priority ranks for nursery school applications. The experiments showed that NCC-UCRL2 outperforms several state-of-the-art bandit algorithms. We study the general NCC bandit problem, where the reward can take any form, linear or nonlinear. Besides, the number of state observations can be arbitrarily large. A potential future research direction would be to allow for restrictive assumptions on the number of state observations or the space of reward functions. In such cases, the dependence of the regret bound on the number of features and partial states diminishes.

REFERENCES

[1] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, MA, USA: Cambridge Univ. Press, Aug. 2020. [Online]. Available: https://tor-lattimore.com/downloads/book/book.pdf

[2] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221006706

[3] Q. Wu, H. Wang, Y. Li, and H. Wang, "Dynamic ensemble of contextual bandits to satisfy users' changing interests," in *Proc. World Wide Web Conf.*, 2019, pp. 2080–2090. [Online]. Available: https://doi.org/10.1145/3308558.3313727

[4] K. Liu and Q. Zhao, "Adaptive shortest-path routing under unknown and stochastically varying link states," in *Proc. IEEE 10th Int. Symp. Model. Optim. Mobile, Ad Hoc Wireless Netw.*, 2012, pp. 232–237.

[5] S. Ghoorchian and S. Maghsudi, "Non-stationary delayed combinatorial semi-bandit with causally related rewards," 2023, *arXiv:2307.09093*.

[6] H. Robbins, "Some aspects of the sequential design of experiments," *Bull. Amer. Math. Soc.*, vol. 58, no. 5, pp. 527–535, 1952.

[7] S. Maghsudi and E. Hossain, "Multi-armed bandits with application to 5G small cells," *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 64–73, Jun. 2016.

[8] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *J. Mach. Learn. Res.*, vol. 11, pp. 1563–1600, 2010.

[9] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *Algorithmic Learning*, J. Theory, C. Kivinen, E. Szepesvári Ukkonen, and T. Zeugmann, Eds. Berlin, Heidelberg: Springer, 2011, pp. 174–188.

[10] H. Luo, C.-Y. Wei, A. Agarwal, and J. Langford, "Efficient contextual bandits in non-stationary worlds," in *Proc. 31st Conf. Learn. Theory*, 2018, pp. 1739–1776. [Online]. Available: https://proceedings.mlr.press/v75/luo18a.html

[11] Y. Chen, C.-W. Lee, H. Luo, and C.-Y. Wei, "A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free," in *Proc. 32nd Conf. Learn. Theory*, 2019, pp. 696–726. [Online]. Available: https://proceedings.mlr.press/v99/chen19b.html

[12] Y. Russac, C. Vernade, and O. Cappé, "Weighted linear bandits for non-stationary environments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/263fc48aae39f219b4c71d9d4bb4aed2-Paper.pdf

[13] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Learning to optimize under non-stationarity," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1079–1087. [Online]. Available: https://proceedings.mlr.press/v89/cheung19b.html

[14] N. Hariri, B. Mobasher, and R. Burke, "Adapting to user preference changes in interactive recommendation," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 4268–4274.

[15] C. Zeng, Q. Wang, S. Mokhtari, and T. Li, "Online context-aware recommendation with time varying multi-armed bandit," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 2025–2034. doi: 10.1145/2939672.2939878.

[16] Q. Wu, N. Iyer, and H. Wang, "Learning contextual bandits in a non-stationary environment," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 495–504. doi: 10.1145/3209978.3210051.

[17] X. Xu, F. Dong, Y. Li, S. He, and X. Li, "Contextual-bandit based personalized recommendation with time-varying user interests," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 6518–6525.

[18] S. Lu et al., "Non-stationary continuum-armed bandits for online hyperparameter optimization," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, 2022, pp. 618–627, doi: 10.1145/3488560.3498396.

[19] K. Kamikokuryo, T. Haga, G. Venture, and V. Hernandez, "Adversarial autoencoder and multi-armed bandit for dynamic difficulty adjustment in immersive virtual reality for rehabilitation: Application to hand movement," *Sensors*, vol. 22, no. 12, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/12/4499

[20] Y. Tang, A. Li, A. A. Scheller-Wolf, and S. R. Tayur, "Multi-armed bandits with endogenous learning and queueing: An application to split liver transplantation," *SSRN Electron. J.*, 2021. [Online]. Available: https://ssrn.com/abstract=3855206

[21] D. E. Losada, J. Parapar, and A. Barreiro, "Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems," *Inf. Process. Manage.*, vol. 53, no. 5, pp. 1005–1025, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457316305660

[22] H. Bastani et al., "Efficient and targeted COVID-19 border testing via reinforcement learning," *Nature*, vol. 599, no. 7883, pp. 108–113, Sep. 2021, doi: 10.1038/s41586-021-04014-z.
[23] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/903ce9225fca3e988c2af215d4e544d3-Paper.pdf
[24] S. Baltaoglu, L. Tong, and Q. Zhao, "Online learning and optimization of markov jump linear models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 2289–2293.
[25] S. Baltaoglu, L. Tong, and Q. Zhao, "Online learning and optimization of markov jump affine models," CoRR, vol. abs/1605.02213, 2016. [Online]. Available: http://arxiv.org/abs/1605.02213
[26] P. Gajane, R. Ortner, and P. Auer, "A sliding-window algorithm for markov decision processes with arbitrarily changing rewards and transitions," CoRR, vol. abs/1805.10066, 2018. [Online]. Available: http://arxiv.org/abs/1805.10066
[27] N. Zolghadr et al., "Online learning with costly features and labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/291597a100aadd814d197af4f4bab3a7-Paper.pdf
[28] J. Janisch, T. Pevny, and V. Lisy, "Classification with costly features as a sequential decision-making problem," *Mach. Learn.*, vol. 109, no. 8, pp. 1587–1615, 2020.
[29] H. Shim, S. J. Hwang, and E. Yang, "Joint active feature acquisition and classification with variable-size set encoding," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1375–1385.
[30] O. Atan, S. Ghoorchian, S. Maghsudi, and M. van der Schaar, "Data-driven online recommender systems with costly information acquisition," *IEEE Trans. Serv. Comput.*, vol. 16, no. 1, pp. 235–245, Jan./Feb. 2023.
[31] Y. Seldin, P. Bartlett, K. Crammer, and Y. Abbasi-Yadkori, "Prediction with limited advice and multiarmed bandits with paid observations," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 280–287. [Online]. Available: https://proceedings.mlr.press/v32/seldin14.html
[32] N. Cesa-Bianchi, S. Shalev-Shwartz, and O. Shamir, "Efficient learning with partially observed attributes," *J. Mach. Learn. Res.*, vol. 12, pp. 2857–2878, 2011.
[33] E. Hazan and T. Koren, "Linear regression with limited observation," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, vol. 1, pp. 1865–1872.
[34] D. Bouneffouf, I. Rish, G. Cecchi, and R. Féraud, "Context attentive bandits: Contextual bandit with restricted context," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1468–1475, doi: 10.24963/ijcai.2017/203.
[35] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*. Berlin, Germany: Springer Science & Business Media, 1991.
[36] R. H. Shumway, D. S. Stoffer, and D. S. Stoffer, *Time Series Analysis and Its Applications*, vol. 3. Berlin, Germany: Springer, 2000.
[37] L. Li, W. Chu, J. Langford, and R. Schapire, "A contextual-bandit approach to personalized news article recommendation," *Comput. Res. Repository - CORR*, 2010.
[38] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2/3, pp. 235–256, 2002.
[39] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

SAEED GHOORCHIAN received the B.Sc. degree in pure mathematics from the Iran University of Science and Technology, Tehran, Iran, in 2014, and the joint M.Sc. degree in applied mathematics from the University of Hamburg, Hamburg, Germany, in 2017. He was a Guest Researcher with the Technical University of Hamburg, Hamburg, in 2018. In 2023, he successfully defended his Ph.D. thesis with the University of Tübingen, Tübingen, Germany. During 2023–2024, he held a Postdoctoral Position with the University of Tübingen and was a Visiting Researcher with SAP. He is currently a Postdoctoral Fellow with the Ruhr-University Bochum, Bochum, Germany. His research interests include machine learning, multi-armed bandits, reinforcement learning, and generative AI.

EVGENII KORTUKOV received the B.Sc. degree in informatics from Moscow State University, Moscow, Russia, in 2020. He is currently working toward the M.Sc. degree in machine learning with the University of Tübingen. His research interests include sequential decision-making and machine learning.

SETAREH MAGHSUDI received the Ph.D. (*summa cum laude*) degree from the Technical University of Berlin, Berlin, Germany, in 2015. From 2015 to 2017, she held Postdoctoral positions with the University of Manitoba, Winnipeg, MB, Canada, and Yale University, New Haven, CT, USA. From 2017 to 2023, she was an Assistant Professor with the Technical University of Berlin and Tübingen University, Tübingen, Germany. She currently a Full Professor with the Ruhr-University of Bochum, Bochum, Germany and a Senior Research Scientist with Fraunhofer Heinrich-Hertz Institute, Berlin. Her research interests include the intersection of network analysis and optimization, game theory, machine learning, and data science. She was the the recipient of several competitive fellowships, awards, and research grants from different institutes, including the German Research Foundation, the German Ministry of Education and Research, and the Japan Society for the Promotion of Science.