POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica e Informazione

Breakpoint Prediction for the Abruptly-Switching Non-Stationary Multi-Armed Bandit Problem

AI & R Lab
Laboratorio di Intelligenza Artificiale
e Robotica del Politecnico di Milano

Relatore: Prof. Marcello Restelli
Correlatore: Francesco Trovò PhD,
Prof. Boracchi,
Dott. Carrera,
Dott. Nuara

Tesi di Laurea di:
Fabio Chiusano, matricola 874294

Anno Accademico 2017-2018

Abstract

Several problems, such as clinical trials and online recommender systems, can be modeled as non-stationary multi-armed bandit (MAB) problems where the distributions of the rewards of the actions change abruptly. Among state-of-the-art policies, actively adaptive policies tackle this problem by coupling policies for the stationary setting with change detection procedures. In this thesis, we propose a novel policy, namely CUSUM-UCB with History (CUH), which takes advantage of the breakpoint prediction capability of change detection procedures. We prove a regret upper bound for an actively adaptive meta-policy which is independent from the stationary subpolicy and the change detection procedure used. Moreover, we prove a regret upper bound for the CUH policy when the maximum amount of history is fixed, which is the first result, up to our knowledge, concerning a MAB policy using breakpoint prediction. Finally, we test CUH against state-of-the-art policies, showing that, when the past history is unbounded, the policy outperforms the state-of-the-art algorithms on the real-world dataset and most of synthetic scenarios.

Sommario

Molti problemi, come i sistemi di raccomandazione online e i test clinici, possono essere modellizzati come problemi di multi-armed bandit (MAB) in cui le distribuzioni delle ricompense cambiano in certi istanti di tempo sconosciuti all'agente. Tra le politiche stato dell'arte, le politiche di adattamento attivo affrontano il problema proponendo un meta-algoritmo che usa congiuntamente una politica per il caso stazionario e una procedura di rilevamento del cambiamento. In questa tesi, proponiamo una nuova politica, chiamata CUSUM-UCB with History (CUH), che utilizza procedure di rilevamento e stima del preciso istante del cambiamento e riutilizza osservazioni precedenti. Dimostriamo un limite superiore sul regret di una meta-politica di adattamento attivo che è indipendente dalla sotto-politica stazionaria e dalla procedura di rilevamento del cambiamento usati. Inoltre, dimostriamo un limite superiore sul regret ottenuto dalla politica CUH quando viene fissato un massimo numero di osservazioni di storia passata, che è il primo risultato di nostra conoscenza riguardante una politica MAB che stima gli istanti dei cambiamenti. Infine, sperimentiamo la nostra politica CUH comparando i risultati con quelli degli stati dell'arte, mostrando come, quando non viene posto un limite massimo sulle osservazioni di storia passata, la nostra politica ottiene risultati migliori rispetto agli stati dell'arte sia su dataset con dati reali, sia sulla maggioranza dei dataset sintetici.

Ringraziamenti

Un enorme ringraziamento va a mia madre Mariangela, mio padre Maurizio, mia sorella Valentina, mia nonna Michelina e agli altri membri della mia grande famiglia, che mi hanno sempre sostenuto e hanno continuato anche durante questo periodo di studio, nonostante ad ogni domanda riguardo all'andamento della tesi ricevessero sempre la stessa esaustiva risposta: "non ne ho idea." Grazie ai vostri sacrifici ho potuto seguire le mie passioni, e non vi ringrazierò mai abbastanza.

Vorrei ringraziare Trovò PhD, Prof. Restelli, Prof. Boracchi, Dott. Carrera e Dott. Nuara per l'insostituibile supporto nella stesura di questa tesi. Avete saputo risolvere ogni mio dubbio con grande competenza, disponibilità e quella simpatia che rende più piacevole lavorare ogni giorno.

Infine, ringrazio i miei amici, con cui ho condiviso momenti di svago e di studio, gioie e preoccupazioni, avventure e disavventure. Grazie per esserci sempre.

Contents

1 Introduction **1**

2 Background Techniques: Multi-Armed Bandit Algorithms and Change Detection Tests **3**

2.1 MAB model . . . 3

2.2 MAB problems taxonomy . . . 4

2.2.1 Stochastic Stationary MAB: formulation and lower bound on expected pseudo-regret . . . 6

2.2.2 Adversarial MAB: formulation and lower bound on expected adversarial regret . . . 8

2.2.3 Other MAB problems . . . 9

2.3 Algorithms . . . 10

2.3.1 Algorithms for stochastic stationary MAB . . . 11

2.3.1.1 ϵ-greedy . . . 11

2.3.1.2 UCB1 . . . 12

2.3.1.3 UCB1-Tuned . . . 14

2.3.1.4 Thompson Sampling . . . 14

2.3.2 Algorithms for adversarial MAB . . . 19

2.3.2.1 Deterministic policies applied to Adversarial MABs . . . 19

2.3.2.2 Exp3 . . . 19

2.4 Change Detection Test procedures . . . 20

2.4.1 CUSUM . . . 21

2.4.2 PHT . . . 23

2.4.3 Breakpoint prediction . . . 25

3 Stochastic Non-stationary MAB problem formulation **29**

3.1 Formulation . . . 29

3.2 Lower bound on pseudo-regret of abruptly changing environments . . . 31

4 Non-stationary MAB state-of-the-art **33**

4.1 D-UCB . . . 34

4.2 SW-UCB . . . 35
4.3 Adapt-EvE . . . 36
4.4 CUSUM-UCB . . . 37

5 Proposed method **43**
5.1 CUSUM-UCB with History: CUH . . . 43
5.2 Analysis . . . 45
5.2.1 Analysis of the meta-algorithm without breakpoint prediction . . . 46
5.2.2 UCB1-EXPLORE with a generic CDT . . . 47
5.2.3 UCB1-EXPLORE with a generic CDT with breakpoint prediction . . . 52
5.2.4 UCB1-EXPLORE with CUSUM . . . 54
5.2.5 Analysis of breakpoint prediction for true alarms with CUSUM . . . 55
5.2.6 Analysis of change instant prediction for false alarms with CUSUM . . . 56
5.2.7 Bias analysis due to breakpoint prediction with CUSUM 56
5.2.8 Analysis of the meta-algorithm with breakpoint prediction . . . 61
5.2.9 Imposing a maximum amount of history given by the breakpoint prediction procedure . . . 62
5.2.10 Expected regret of CUH . . . 62
5.3 Discussion . . . 63

6 Experiments **65**
6.1 Experimental setting . . . 65
6.2 Results . . . 70
6.2.1 Scenario 1: optimal arm changes versus sub-optimal arm changes . . . 70
6.2.2 Scenario 2: optimal arm always changes . . . 74
6.2.3 Scenario 3: optimal arm does not change . . . 79
6.2.4 Scenario 4: bounded changes . . . 80
6.2.5 Scenario 5: real-world data with Yahoo! . . . 82

7 Conclusion **85**

Chapter 1

Introduction

Consider the situation in which a gambler is in front of a row of slot machines (a.k.a. one-armed bandit) and must decide which one to play. Each machine has a different winning probability, meaning that the gambler can maximize his earnings by playing the arm with highest winning probability. However, no prior information is given. For this reason, the gambler initially may want to spend her/his money trying different slot machines and record all the rewards she/he received. As soon as there is enough evidence that an option is better than another, the gambler will only play consistently the best identified arm, with the purpose of maximizing the total reward he/she gained during this process. This and many other real-world problems can be modeled as an online decision making problem, and, more specifically, a Multi-Armed Bandit (MAB) problem. This framework has been introduced by [1] and takes its name from the aforementioned gambling example. The problem concerns an agent whose goals is to maximize its reward (or equivalently to minimize the loss) gained during the process of learning the option providing the largest reward. In such a problem the learner faces the problem of balancing between two different kind of choices: choose between selecting an explorative action, which improves the information she/he has on the problem, or an exploitative one, which tries to maximize the reward she/he obtains given the information gathered so far. This dilemma is known in the MAB literature as the *exploration/exploitation dilemma*. A MAB policy must find the optimal trade-off between exploration and exploitation, to maximize the total reward the learner obtains over a finite time horizon.

There are many real-world applications of the MAB model. An example is the problem of clinical trials, which consists in investigating the effect of different experimental treatments, while minimizing patient losses [1]. In online recommendation systems [2], the goal is to suggest different items in order to understand the tastes of the client, which is an explorative action, while maximizing the interest of the client for the items proposed, which is an exploitative action, thus being suitable to be formulated with the MAB

model. Firms may decide among different multiple versions of their ads in order to learn which one is the most effective [3]. MAB algorithms handle this problem automatically, providing the optimal trade-off between exploration of different ads and exploitation of the best one. A further example is the task of sending packets between two nodes in a communication network [4]. There are several possible routes and the learner must choose one for each packet to minimize the transmission cost.

However, there are problems where the best action changes as time goes on. For instance, during clinical trials, as a disease evolves, the effect of treatments may change. Clients may take different interests as time passes and recommender systems should be able to react. These scenarios encourage the development of ad-hoc strategies that take into account the non-stationarities of the problem. Several policies have been developed and studied [5] [6], which actively look for the time instants in which the best arm changes through statistical procedures of change detection [7]. Yet, existing policies does not make use of the change detection procedures capability of identifying the correct instant in which a change happened. Indeed, the use of this information might speed up the learning process, therefore increasing the total reward gained during the online learning procedure.

In this thesis, we propose a novel policy, namely CUH, that employs breakpoint prediction, which is able to handle situations in which the change is abrupt. We provide a theoretical analysis, to bound the loss one might incur while using the proposed algorithm, and perform a wide experimental campaign to compare what has been proposed with state-of-the-art policies for this specific problem.

The structure of this thesis is as follows: in Chapter 2, the stationary and adversarial variants of the MAB problem are formulated and we present policies to deal with them; the non-stationary MAB problem is formulated in Chapter 3 and state-of-the-art algorithms for this setting are presented in Chapter 4; a novel policy which incorporates the prediction of breakpoints and the use of past history is described, analysed and discussed in Chapter 5; the results of simulations concerning our novel policy and state-of-the-art algorithms are discussed in Chapter 6. Finally, in Chapter 7, we draw conclusions on the work done and we draw future lines of research.

Chapter 2

Background Techniques: Multi-Armed Bandit Algorithms and Change Detection Tests

In this chapter, we formulate the stationary and adversarial MAB problems, we explain how much a policy can perform well on these settings, and we present some policies that deal with them. Finally, we introduce change detection test procedures, as they are later employed by state-of-the-art policies for the non-stationary setting.

2.1 MAB model

A MAB is formally equivalent to a one-step Markov Decision Process (MDP) [1]:

Definition 1. (MAB) A **MAB** problem is a tuple $< \mathcal{K}, R >$ where:

- $\mathcal{K}$ is the set of K possible actions, called arms;
- $R := R(r|a)$ is an unknown probability distribution of rewards r given the chosen action a.

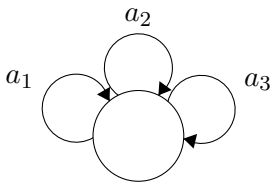


Figure 2.1: Formulation of a MAB problem with 3 arms. When the agent selects an action, she/he receives a reward that depends on the chosen action and returns to the initial state.

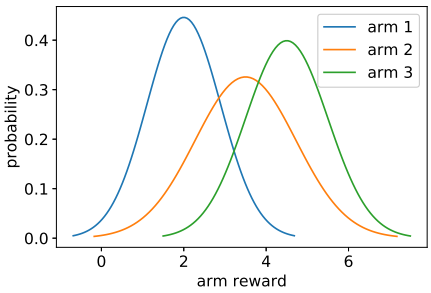


Figure 2.2: Graphical illustration of the probability distributions associated to the arms.

Arm	*Mean*	*Variance*
a_1	2	0.8
a_2	3.5	1.5
a_3	4.5	1

Table 2.1: Parameters of the probability distributions associated to the arms.

We denote with K the cardinality of the arm set, or, formally, $K := |\mathcal{K}|$.

Example 1. In Figure 2.1 we show an example of a MAB represented as a one-step MDP with $\mathcal{K} = \{a_1, a_2, a_3\}$. The reward distributions $R(r|a)$ associated to the arms are Gaussian in this specific example. This setting is described in Figure 2.2 and table 2.1. In this example, the arm with highest mean reward is a_1. However, taking two random samples of arms a_1 and a_2, it may happen that the reward of a_1 is smaller than the reward of a_2, even though their average rewards shows the opposite behaviour. For this reason, each arm must be sampled a sufficient number of times to be confident in saying that "arm a_i has larger mean reward that arm a_j".

2.2 MAB problems taxonomy

There are different variants of the MAB model. The taxonomy of MAB variants is summarized in Figure 2.3. Among the most important characteristics of a MAB, we have that a specific problem can be:

- *Stochastic* vs *Adversarial*: in a Stochastic Multi-Armed Bandit (S-MAB) [8], the rewards are generated from stochastic distributions. In an Adversarial Multi-Armed Bandit (A-MAB) [9], the rewards are generated by a process that cannot be treated as a stochastic distribution. In this latter setting, we say that the rewards are generated by an adversary, who may take advantage of those corner cases in which a bandit algorithm performs badly. As a consequence, adversarial algorithms must be robust to adversary choice of the rewards;

- *Stochastic Stationary* vs *Stochastic Non-Stationary*: in a Stationary Multi-Armed Bandit (ST-MAB), the rewards are generated from stochastic distributions which are stationary, i.e., they do not change as time

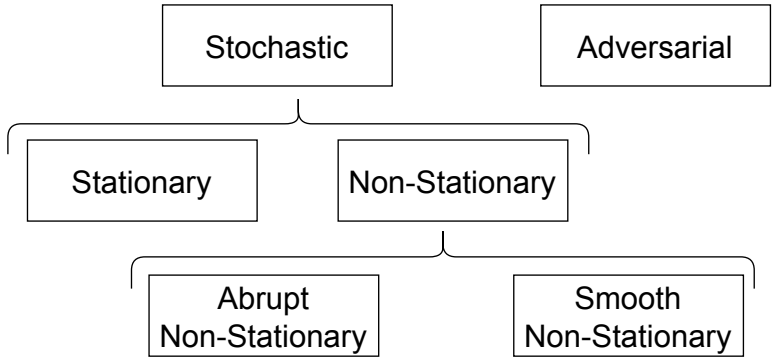


Figure 2.3: Taxonomy of the MAB settings.

goes on. In a Non-Stationary Multi-Armed Bandit (NS-MAB) [10], the stochastic distributions may change at each round.

Moreover, NS-MAB settings in which the mean reward of the arms changes at each round without further constraints are intractable. Among them, we make this further distinction:

- *Abrupt changes* vs *Smooth changes*: we call Abruptly changing Non-Stationary Multi-Armed Bandit (ANS-MAB) the setting in which the distributions change a number of times which is a function of the time horizon, but the magnitude of the change is unbounded. We call Smoothly changing Non-Stationary Multi-Armed Bandit (SNS-MAB) the setting in which the parameters of the distributions smoothly change at each timestep by less than a certain value.

This taxonomy is useful as it allows to correctly formalize a problem and solve it with the right techniques. Indeed, it is often unrealistic to model the real world with stationary distributions and many problems can be modelled as non-stationary or adversarial MAB problems. We now categorize the applications of the MAB model presented in Chapter 1. The clinical trials problem, in its first formulation [1] was considered as a stationary MAB. However, as a disease evolve, the effect of treatments may change and, thus, it may be handled by SNS-MAB techniques. In online recommendation systems [2], the algorithm must be able to adapt to the user having a particular interest on one day, and, thus, it may be modeled as a ANS-MAB. On the other hand, consider the task of sending packets between two nodes in a communication network [4], in which there are K possible routes and the decision maker must choose one for each packet in order to minimize the transmission cost. It is likely that the costs associated with each route cannot be modeled with stochastic distribution, thus this is an A-MAB setting. Moreover, parameters control [11], investments [12] or the dynamic spectrum access problem [13], can be modelled as non-stationary problems. This thesis

focuses on ANS-MAB problems. SNS-MAB settings, where the distribution of rewards change continuously, are considered in [14].

2.2.1 Stochastic Stationary MAB: formulation and lower bound on expected pseudo-regret

In this section, we formulate the stationary MAB problem and we introduce the difference between regret and pseudo-regret. Next, we give a lower bound on the expected pseudo-regret achievable in this problem.

A stationary MAB is a stochastic MAB whose arms are characterised by stationary distributions. As a consequence, everything in this section relies upon this assumption:

Assumption 1. The rewards of the arms can be modeled with i.i.d. random variables such that:

$$\forall i \in \mathcal{K}, \forall t \in \{1, \dots, T-1\} | \mu_t(i) = \mu_{t+1}(i) = \mu(i),$$

where $\mu(i)$ is the average reward of arm i.

The goal of a bandit algorithm $\mathfrak{U}$ is to maximize the sum of rewards received over T time slots, which is equivalent to minimizing the loss incurred during the learning process. Let $X_{i,t}$ be the reward obtained by pulling arm i at round t and i_t be the arm selected at round t by the policy $\mathfrak{U}$.

Definition 2. (Regret) The **regret** $L_T(\mathfrak{U})$ of a MAB policy $\mathfrak{U}$ at time T is the sum, up to the time horizon T, of the differences between the rewards of the arm that would return the highest expected reward if always pulled, and the rewards of the arms selected by the policy $\mathfrak{U}$, or, formally:

$$L_T(\mathfrak{U}) = \max_{i \in \mathcal{K}} \sum_{t=1}^{T} X_{i,t} - \sum_{t=1}^{T} X_{i_t,t}$$

Thus, we obtain the **expected regret** by taking the expected value over the experiments:

$$\mathbb{E}[L_T(\mathfrak{U})] = \mathbb{E}\left[\max_{i \in \mathcal{K}} \sum_{t=1}^{T} X_{i,t} - \sum_{t=1}^{T} X_{i_t,t}\right]$$

A different quantity is the pseudo-regret, in which we consider the expectation over the stochasticity of the reward process:

Definition 3. (Pseudo-regret) The **pseudo-regret** $\bar{L}_T(\mathfrak{U})$ of a MAB policy $\mathfrak{U}$ at time T is the sum, up to the time horizon T, of the differences between the mean of the optimal arm $i^* = \arg\max_i \mu_i$ and the means of the arms selected by the policy $\mathfrak{U}$.

$$\bar{L}_T(\mathfrak{U}) = \sum_{t=1}^{T} (\mu_{i^*} - \mu_{i_t}) = T\mu_{i^*} - \sum_{t=1}^{T} \mu_{i_t}$$

Thus, we obtain the **expected pseudo-regret** by taking the expected value over the experiments:

$$\mathbb{E}[\bar{L}_T(\mathfrak{U})] = \mathbb{E}\left[\sum_{t=1}^{T}(\mu_{i^*} - \mu_{i_t})\right] = \mathbb{E}\left[T\mu_{i^*} - \sum_{t=1}^{T}\mu_{i_t}\right]$$

Example 2. Using the same setting described in Table 2.1, if $T = 3$, assume that the arms realizations have been $a_1 \rightarrow \{1.8, 2.1, 2.2\}$, $a_2 \rightarrow \{4.1, 3.8, 4.4\}$ and $a_3 \rightarrow \{4.0, 3.9, 4.2\}$. The optimal arm, with respect to the pseudo-regret, is a_3 because $\mu_3 = \max(\mu_1, \mu_2, \mu_3)$. The optimal arm, with respect to the regret, is a_2 because $\sum_{t=1}^{T} X_{2,t} = \max(\sum_{t=1}^{T} X_{1,t}, \sum_{t=1}^{T} X_{2,t}, \sum_{t=1}^{T} X_{3,t})$. If our policy samples the arms $\{i_1 = a_1, i_2 = a_2, i_3 = a_3\}$, its regret is $\sum_{t=1}^{T} X_{2,t} - (X_{1,1} + X_{2,2} + X_{3,3}) = 2.5$ and its pseudo-regret is $3\mu_3 - (\mu_1 + \mu_2 + \mu_3) = 3.5$.

It is possible to rewrite the expected pseudo-regret definition in the following way:

$$\begin{aligned}\mathbb{E}\left[\bar{L}_T(\mathfrak{U})\right] &= \sum_{i \in \mathcal{K}} \left(\mathbb{E}\left[N_T(i)\right](\mu_{i^*} - \mu_i)\right) \\ &= \sum_{i \in \mathcal{K}} \left(\mathbb{E}\left[N_T(i)\right](\Delta_i)\right)\end{aligned}$$

where

$$N_T(i) = \sum_{t=1}^{T} \mathbb{1}_{\{i_t = i\}}$$

$$\Delta_i = \mu_{i^*} - \mu_i$$

With this last formulation, it is easy to see that the expected pseudo-regret is a function of:

- $\mathbb{E}\left[N_T(i)\right]$: the average number of times arm i has been pulled up to time step T;
- Δ_i: the difference between the mean reward of the best arm and the mean reward of arm i.

Notice that, in this formulation, we are not taking into account possible non-stationarities in the distributions, since the optimal arm remains the same up to the time horizon T. In Chapter 3, we will see also the corresponding formulation of the NS-MAB, where the optimal arm changes over time.

It turns out that every policy that operates on a stochastic stationary MAB achieves at least a logarithmic expected pseudo-regret as $T \to \infty$. Indeed:

Theorem 2.1. *(ST-MAB problem lower bound on pseudo-regret [15]) Given a ST-MAB problem, any policy $\mathfrak{U}$ has the following lower bound on the pseudo-regret:*

$$\lim_{T\to\infty} \mathbb{E}\left[\bar{L}_T(\mathfrak{U})\right] \geq \log T \sum_{i|\Delta_i>0} \frac{\Delta_i}{KL(R(i), R(i^*))},$$

where $R(i)$ is the reward distribution of arm i, i^* is the arm with the highest mean reward and $KL(R(i), R(i^*))$ is the Kullback-Leibler divergence between the two distributions $R(i)$ and $R(i^*)$.

The Kullback-Leibler divergence represents a dissimilarity between two distributions [16]. In the simple case, a value close to zero indicates that we can expect a very similar behaviour from the two distributions. On the contrary, a value close to one indicates that the two distributions behave in a different manner. Theorem 2.1 also shows that:

- large gaps Δ_i between the means imply a higher total pseudo-regret, since the exploration rounds cost a lot if gaps are large;
- similar distributions $R(i)$ and $R(j)$ with $KL(R(i), R(j)) << 1$ lead to larger total pseudo-regret, since it is more difficult to say that they are significantly different.

Several algorithms have been proposed and proven to achieve $O(\log(T))$ expected pseudo-regret, such as Upper Confidence Bound 1 (UCB1) and Thompson Sampling (TS). Notice that this is a distribution dependent bound, i.e., it depends on Δ_i. When $\Delta_i \to 0$, this lower bound increases to infinity. However, it is clear that the pseudo-regret incurred from always pulling arm i cannot be greater than $T\Delta_i$, thus, policies have been developed to achieve low pseudo-regret even with very small Δ_i [17].

2.2.2 Adversarial MAB: formulation and lower bound on expected adversarial regret

In this section, we formulate the adversarial MAB problem, we introduce the notion of adversarial regret and we give a lower bound on the expected adversarial regret achievable on such a problem.

Definition 4. (Adversarial MAB [9]) An **adversarial MAB** problem is specified by the number of possible actions K and a reward vector $X_t = (X_{1,t}, \ldots, X_{K,t})$ for each round t, where $X_{i,t} \in [0, 1]$.

In an A-MAB, the rewards are generated by a process that cannot be considered stochastic. For instance, an adversary may generate ad-hoc rewards that make our policy achieve as much regret as possible. As a consequence, deterministic algorithms cannot be used (e.g., UCB1), because the adversary would always know the chosen arm and would assign it a low reward.

The A-MAB problem can be formalised as a game between a player choosing an action and a player choosing the rewards. At each timestep t, the following happens:

1. The adversary selects a reward for each arm.
2. Without knowledge of the choices of the adversary, the player selects an arm and observe its reward.

Since there are no more stochastic distributions associated to the arms and, thus, it is impossible to compute the mean reward of each arm, it is not possible anymore to use the stochastic regret definition, which is substituted by the notion of weak regret (also known as adversarial regret):

Definition 5. (Adversarial regret) The **weak regret** or **adversarial regret** $L_T(\mathfrak{U})$ [9] achieved by policy $\mathfrak{U}$ on a A-MAB is defined as:

$$L_T(\mathfrak{U}) = \max_i \left\{ \sum_{t=1}^{T} X_{i_t,t} \right\} - \sum_{t=1}^{T} X_{i,t}$$

where $X_{i,t}$ is the reward of arm i at time t and $X_{i_t,t}$ is the reward of the arm selected by the policy at time t.

Therefore, the weak regret compares the actions chosen by the policy with the best constant action. The following theorem states that the expected adversarial regret of any policy that acts in an A-MAB is at least $O\left(\sqrt{KT}\right)$.

Theorem 2.2. *(Lower bound on worst-case adversarial regret [9]) For any policy for the A-MAB problem, there exists a sequence of reward vectors such that the expected adversarial regret of the policy playing in such setting over T timesteps is lower bounded by $\Omega\left(\sqrt{KT}\right)$.*

This bound is weaker than the one found for stochastic bandits. However it is valid for each finite time horizon T, while the bound on stochastic bandits is valid for $T \to \infty$: this is the main reason of the gap between the two bounds, not the adversarial nature [9].

2.2.3 Other MAB problems

As stated in Section 2.2, there are several variants of the MAB model. Stochastic MAB are characterised by arms that can be described with probability distributions, while Adversarial MAB not. These distributions can

be stationary or non-stationary: in the latter case, changes might occur at each round but limited in magnitude (i.e., smooth), or might occur only on a limited number of rounds but not limited in magnitude (i.e., abrupt).

Among the others, we cite the most relevant ones. Contextual bandits consider also a context (i.e., a vector of features associated to a round) associated to each interaction with the environment: the decision maker must decide not only with the knowledge of past rewards, but also with other features associated to each reward. An example of usage of the contextual bandit model is to recommend personalized news article, where the context is a set of article features and user features [18].

Linear bandits relax the hypothesis of a finite discrete set of arms to a compact set. In this case, the loss is a linear function defined on such compact set and the goal is to pick an arm as close as possible to the minimum of such function [19]. Non-linear bandits extend linear bandits with non-linear losses.

Bandits may change according to the feedback structure. Full-information bandit [20] assumes that the regret is computed on the arm chosen by the policy, but the policy observes a reward for each action. Instead, in partial-monitoring bandits [21] a policy must choose only a subset of arms to observe at each timestep.

Summing up, variants of the MAB model are developed among several axes, such as [22]:

- evolution of payoffs over time: e.g., stochastic, adversarial, non-stationary;
- structure of payoff functions: e.g., linear, non-linear, gaussian process;
- feedback structure: e.g., full-information, bandit, partial-monitoring;
- presence of context;
- type of regret.

2.3 Algorithms

In this section, we introduce policies that deal with the stochastic stationary and the adversarial settings. In Chapter 4 we present the state-of-the-art policies for the stochastic non-stationary case, being the focus of this thesis.

Stochastic MAB problems, characterised by the presence of stochastic distributions, can be handled with three different reasoning methods:

- Frequentist reasoning: the parameters of the distribution of the observed rewards are scalar unknown parameters. The policy selects an arm at each time step based on the observed history. Some examples are the UCB1 [8] and Upper Confidence Bound 1 Tuned (UCB1-Tuned) [23] policies.

- Bayesian reasoning: the parameters of the distributions of the observed rewards are random variables with a prior distribution. The policy selects an arm at each round based on the posterior distribution, built using both the observed history and on the provided prior, which is updated as samples are received. An example is the TS policy [24].
- Hybrid frequentist and bayesian: there exists policies that combines both reasonings, for example BayesUCB [25].

2.3.1 Algorithms for stochastic stationary MAB

In this section, we present several policies that deal with stochastic stationary MAB problems.

2.3.1.1 ϵ-greedy

A possible solution to the MAB problem is to use an ϵ-greedy policy, which is a widely-used simple heuristic used for exploration easily generalisable for online decision problems [26]. At each round, the policy selects the arm with highest empirical mean with probability $1-\epsilon$, otherwise it chooses a random arm. Thus, the probability of choosing arm i at round t is:

$$p_t(i) = \begin{cases} 1-\epsilon, & \text{if } \hat{X}_t(i) = \max_{j \in \mathcal{K}} \hat{X}_t(j) \\ \frac{\epsilon}{K}, & \text{otherwise} \end{cases}$$

where $\hat{X}_t(i)$ is the sample average of the rewards obtained from the policy by arm i up to time t, formally:

$$\hat{X}_t(i) = \frac{1}{N_t(i)} \sum_{s=1}^{t} (X_{i,s} \mathbb{1}_{i_s=i})$$

$\epsilon \in [0,1]$ is a parameter of the policy and it is called *exploration parameter*.

Using a constant ϵ, a suboptimal arm i is pulled on average $T\frac{\epsilon}{K}$ times up to time T, and the average gap between the optimal arm and a random suboptimal arm is $\frac{1}{K}\sum_{i \in \mathcal{K}} [\Delta_i]$. As a consequence, the ϵ-greedy policy has the following lower bound on the expected pseudo-regret:

$$\mathbb{E}\left[\bar{L}_T(\mathfrak{U}_{\epsilon-greedy})\right] \geq \left(T\frac{\epsilon}{K}\right)\left(\frac{1}{K}\sum_{i \in \mathcal{K}} [\Delta_i]\right)$$

Poly-logarithmic bounds for variants of the algorithm in which ϵ decreases over time have been shown [8]. However, later empirical studies revealed no advantages in this approach with respect to standard ϵ-greedy policies [27]. As a conclusion, the ϵ-greedy policy achieves a linear expected pseudo-regret, i.e., linearly proportional with the time horizon T.

2.3.1.2 UCB1

UCB is a suite of stationary MAB algorithms that use the frequentist approach [8]. Their strategy to cope with the exploration versus exploitation problem is to rely on the so-called *optimism in face of uncertainty* [15]: when the expected reward of an arm is uncertain and the probability of it being the optimal action is high enough, the policy favours the selection of that arm. As more samples are gathered, the estimates of the arms rewards become more and more accurate, lessening the effect of optimism and eventually choosing the action with highest mean reward. Following this heuristic, UCB policies consider an upper bound $\hat{U}_t(i)$ over the empirical mean $\hat{X}_t(i)$ of arm i and, at each time step, select the arm with the highest upper bound.

The bound used by such policies is designed such that, with high probability:

$$\hat{U}_t(i) := \hat{X}_t(i) + \hat{B}_t(i) \geq \mu_i$$

that is, the upper bound on the mean of arm i must be higher than the true mean of arm i with high probability.

What is still left to define is $\hat{B}_t(i)$. In general, we want that:

- small $N_t(i) \rightarrow$ large $\hat{U}_t(i)$, because the estimated value $\hat{X}_t(i)$ is uncertain;
- large $N_t(i) \rightarrow$ small $\hat{U}_t(i)$, because the estimated value $\hat{X}_t(i)$ is accurate;

What differentiates the UCB algorithms is the way in which the upper bounds are computed. UCB1 does this with the Hoeffding inequality [28]:

Definition 6. (Hoeffding bound) Let $X_1, ..., X_t$ be random variables with support in $[0, 1]$ and identical mean $\mathbb{E}[X_i] = \mu$ and let $\hat{X}_t = \frac{1}{t}\sum_{i=1}^{t} X_i$ be the sample mean of the random variables. Then:

$$P\left(\mu > \hat{X}_t + u\right) \leq e^{-2tu^2}.$$

In the case of UCB1, the Hoeffding inequality is applied to every arm. $X_1, ..., X_t$ are the pulls done on the considered arm, and u can be modified to model the precision that we want to achieve on such arm. Applying the Hoeffding bound on a specific arm i, we obtain:

$$\underbrace{P\left(\mu_i > \hat{X}_{N_t(i)}(i) + \hat{B}_t(i)\right)}_{p} \leq e^{-2N_t(i)\hat{B}_t(i)^2}$$

Solving for $\hat{B}_t(i)$ in function of p:

$$\ln(p) \leq -2N_t(i)\hat{B}_t(i)^2$$

and eventually

$$\hat{B}_t(i) \geq \sqrt{\frac{-\ln(p)}{2N_t(i)}}$$

To converge to the optimal solution, we want p to decrease over time, e.g.:

$$p = t^{-4}$$

and we obtain the final upper bound formula of UCB1:

$$\hat{B}_t(i) \geq \sqrt{\frac{-\ln(t^{-4})}{2N_t(i)}} = \sqrt{\frac{2\ln(t)}{N_t(i)}}.$$

The pseudocode of the UCB1 algorithm is presented in Algorithm 1.

Algorithm 1 UCB1

1: for t from 1 to K, play arm $i_t = t$
2: for t from $K+1$ to T, play arm

$$i_t = \arg\max_{i \in \mathcal{K}} \{\hat{X}_t(i) + \hat{B}_t(i)\}$$

Example 3. An example of the execution of the UCB1 algorithm is represented in Figure 2.4. Consider a MAB problem with three gaussian arms. In the first three rounds, UCB1 tries all the arms, and receives rewards 3 for a_1, 5 for a_2 and 4 for a_3. Arm a_2, using the Hoeffding bound, has the highest upper bound, therefore UCB1 selects a_2. Suppose that a_2 returns 3.4 this time. Even though arm a_2 still has the highest sample average $\hat{X}_5(a_2)$ among the arms, a_3 has the highest upper bound $\hat{U}_5(a_3)$. This happens because:

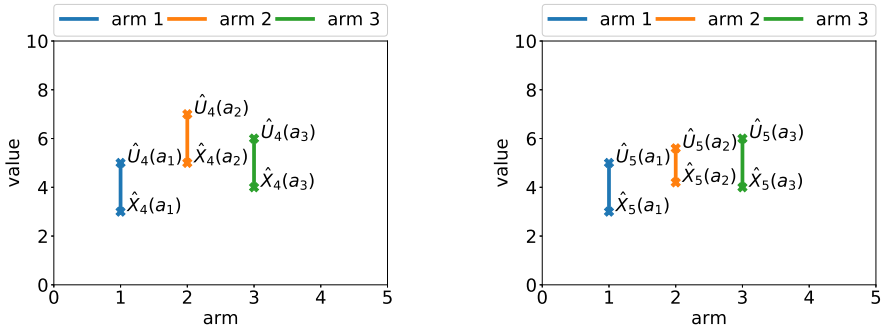
- the sample average of a_2 comes from more observations than that of the other arms, and for this reason $\hat{B}_4(a_2)$ should be smaller;
- as time passes, $\hat{B}_t$ of the least explored arms grows faster than that of the most explored arm.

UCB1 will choose a_3 at round 5.

UCB1 has a $O(\ln(T))$ expected pseudo-regret, as stated by the following theorem:

Theorem 2.3. *(UCB1 regret upper bound [8]) At time T, the expected total regret of the UCB1 policy applied to a ST-MAB problem is:*

$$\mathbb{E}[L_T(\mathfrak{U}_{UCB1})] \leq 8\ln(T) \sum_{i|\Delta_i>0} \frac{1}{\Delta_i} + \left(1 + \frac{\pi^2}{3}\right) \sum_{i|\Delta_i>0} \Delta_i.$$



(a) (b)

Figure 2.4: Example of a UCB1 run on a ST-MAB setting with three arms, with $t = 4$ in (a) and $t = 5$ at (b). The next arms that the UCB1 policy chooses are a_2 in (a) and a_3 in (b).

This means that the algorithm achieves asymptotically optimal expected pseudo-regret. Better bounds have been proven for modifications of the UCB1 algorithm, for example Improved UCB [17] and Kullback Leibler Upper Confidence Bound (KL-UCB) [29].

2.3.1.3 UCB1-Tuned

UCB1-Tuned [30] is similar to UCB1, except that it takes into account also the estimated variance among samples of the same arm and is tuned for Bernoulli MAB, i.e., problems where all the actions have Bernoulli distributions. The idea is that, if the samples from an arm have small variance, we can consider the mean estimate to be more precise and reduce the exploration on such arm.

No theoretical guarantees have been proven for the UCB1-Tuned policy, but has been shown to outperform UCB1 in most scenarios [31]. The pseudocode of UCB1-Tuned is equal to the one of UCB1, except for the use of an upper confidence bound defined as follows

$$\hat{B}_t(i) = \sqrt{\frac{2\ln(t)\min\{\frac{1}{4}, Var_t(i))\}}{N_t(i)}}$$

where $\frac{1}{4}$ is an upper bound for the variance of a Bernoulli variable and $Var_t(i)$ is the sample variance at round t of arm i.

2.3.1.4 Thompson Sampling

Thompson Sampling [24] is a MAB policy which uses the bayesian approach. It needs a bayesian prior for each arm as a starting point, which encodes the initial belief that a learner has on each arm reward. Subsequently, at each

round t, it samples from each arm distribution, chooses the action with highest sample and updates its distribution using the Bayes's rule with the received reward. The updated distribution is called the posterior distribution.

The Bayes's rule states the following:

$$P(H|D) = \frac{P(H)P(D|H)}{P(D)}$$

In the formula we can distinguish:

- $P(H)$ is the prior distribution;
- $P(D|H)$ is the probability of generating data D from hypothesis H, also called likelihood;
- $P(H|D)$ is the posterior distribution, that is the prior updated with data D;
- $P(D)$ is the normalizing constant, which is needed in order to obtain a probability distribution as posterior.

In general, it is not easy to compute the posterior distribution, except for specific choices of the prior distribution and the likelihood function such that the prior and the posterior distributions are from the same family: the prior and the posterior are then called conjugate distributions, and the prior is called conjugate prior for the likelihood function. For example, the beta family is conjugate to itself with respect to a Bernoulli likelihood, and the gaussian family is conjugate to itself with respect to a gaussian likelihood. We show how to develop a TS policy for both of these cases.

We consider now the simple case of a MAB with Bernoulli rewards, i.e. arms that provide a reward of value one with probability μ and zero otherwise. As previously stated, if the prior is a Beta distribution and the Bernoulli samples are i.i.d. (i.e., independent and identically distributed), then the posterior distribution is a Beta distribution too. The Beta distribution is defined as follows:

Definition 7. (Beta distribution [32]) A **Beta distribution** has support $(0, 1)$ with two parameters (α, β) with probability density function:

$$f(x|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1 - x)^{\beta-1},$$

where Γ is the Gamma function. For integers $x \geq 1$, $\Gamma(x) = (x - 1)!$. If x is a random variable distributed from the Beta distribution with parameters (α, β), we have $\mathbb{E}[x] = \frac{\alpha}{\alpha+\beta}$.

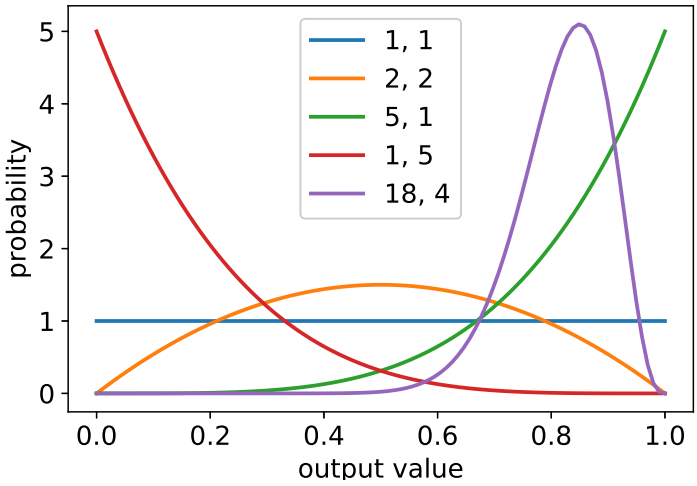


Figure 2.5: Examples of Beta distributions with different parameters α and β. The first value shown in the legend is α, the second is β. When $\alpha = \beta = 1$, the distribution is the uniform distribution over the domain $[0, 1]$: for this reason, this configuration is used as uninformative prior by the TS policy for Bernoulli distributions. Notice that, the bigger is α (respectively, β) with respect to β (α), more the distribution is concentrated towards 1 (0).

The Beta posterior distribution can be computed in the following way, where r is the new reward received:

$$\begin{aligned}
\underbrace{P(\mu = \theta | r)}_{\text{posterior}} &\propto \underbrace{P(r|\mu = \theta)}_{\text{likelihood}} \underbrace{P(\mu = \theta)}_{\text{prior}} \\
&= Bernoulli_\theta(r) Beta_{\alpha,\beta}(r) \\
&= \theta^r (1-\theta)^{1-r} \frac{\Gamma(\alpha, \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1} \\
&= \frac{\Gamma(\alpha, \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha+r-1} (1-\theta)^{\beta-r} \\
&\propto Beta_{\alpha+r,\beta+1-r}(\theta)
\end{aligned}$$

Therefore, starting with a Beta with parameters $\alpha = 1$ and $\beta = 1$, i.e., a uniform distribution over $[0, 1]$, after observing n i.i.d. samples from a Bernoulli distribution, the posterior is a $Beta(1+n_1, 1+n_0)$, where n_1 is the number of successes (positive rewards) in the n samples and n_0 is the number of failures. Thus, when a new sample arrives, the policy just updates α or β depending on the new sample being a one or a zero. The TS algorithm for Bernoulli ST-MAB with Beta priors is defined in Algorithm 2.

Algorithm 2 Thompson Sampling algorithm for Bernoulli MAB with Beta priors

for t=1,2,...,T **do**
  for each arm $i = 1, \ldots, K$ **do**
    Independently sample $\theta_{i,t} \sim Beta(1 + n_1, 1 + n_0)$
  end for
  Play arm $i_t = \arg\max_i \theta_{i,t}$
  Observe reward $r_{i,t}$ and update distribution
end for

This policy has the following problem-dependent upper bound on the expected pseudo-regret:

Theorem 2.4. *(Thompson Sampling problem dependent regret upper bound [24]) The expected pseudo-regret of the TS policy is upper bounded by:*

$$\mathbb{E}\left[\bar{L}_T(\mathfrak{U}_{TS})\right] \leq O\left(\sum_{i \neq i^*} \frac{\Delta_i}{KL(\mu_i, \mu^*)}(\ln(T) + \ln(\ln(T)))\right).$$

Let Θ be the set of means of the Bernoulli arm distributions. A problem-independent upper bound on the expected pseudo-regret is the following:

Theorem 2.5. *(Thompson Sampling problem independent regret upper bound [24]) For any instance of Bernoulli MAB, the expected pseudo-regret of the TS policy is upper bounded by:*

$$\mathbb{E}\left[\bar{L}_T(\mathfrak{U}_{TS})\right] \leq \max_{\Theta} L_{T,\Theta}(\mathfrak{U}_{TS}) \leq O\left(\sqrt{KT\log(T)} + K\right).$$

Now, we consider the case in which the observations are generated from a Gaussian $\mathcal{N}(\mu, 1)$, where μ is an unknown parameter that we want the policy to learn. Starting with a $\mathcal{N}(0, 1)$ prior, we prove that, after n i.i.d. samples from $\mathcal{N}(\mu, 1)$, the posterior distribution for the parameter μ is $\mathcal{N}(\hat{\mu}_n, \frac{1}{n+1})$, where $\hat{\mu}_n$ is the empirical average of the n samples. $\hat{\mu}_0 = 0$ by definition. This can be proven by induction:

1. base case: $\mathcal{N}(\hat{\mu}_n, \frac{1}{n+1}) = \mathcal{N}(0, 1)$ since $\hat{\mu}_0 = 0$ and $n = 0$.

2. inductive hypothesis: at step n, the posterior is $\mathcal{N}(\hat{\mu}_n, \frac{1}{n+1})$. Let r be

the new reward received.

$$\begin{aligned}
\underbrace{P(\theta|r)}_{\text{posterior}} &\propto \underbrace{P(r|\theta)}_{\text{likelihood}}\ \underbrace{P(\theta)}_{\text{prior}} \\
&\propto e^{-\frac{1}{2}(r-\theta)^2} e^{-\frac{n+1}{2}(\theta-\hat{\mu}_n)^2} \\
&\propto e^{-\frac{1}{2}(\theta^2(n+2)-2\theta(r+\hat{\mu}_n(n+1)))} \\
&= e^{-\frac{1}{2}(\theta^2(n+2)-2\theta\hat{\mu}_{n+1}(n+2))} \\
&\propto e^{-\frac{1}{2}(\theta^2(n+2)-2\theta\hat{\mu}_{n+1}(n+2)+\hat{\mu}_{n+1}^2(n+2))} \\
&\propto e^{-\frac{n+2}{2}(\theta-\hat{\mu}_{n+1})^2}
\end{aligned}$$

which is proportional to the probability density function of $\mathcal{N}(\hat{\mu}_{n+1}, \frac{1}{n+2})$.

The TS algorithm for Gaussian ST-MAB with Gaussian priors is defined in Algorithm 3.

Algorithm 3 Thompson Sampling algorithm for Gaussian MAB

```
for t=1,2,...,T do
    for each arm i = 1, ..., K do
        Independently sample θ_{i,t} ~ N(μ̂_{i,t-1}, 1/(n_{i,t-1}+1))
    end for
    Play arm i_t = arg max_i θ_{i,t}
    Observe r_{i,t} and update distribution
end for
```

In general, it may be not possible to assume an arm distribution to be Gaussian or Bernoulli. However, algorithm 3 is still valid even if rewards do not have a Gaussian distribution. Strictly speaking, this algorithm would not be a Bayesian posterior sampling algorithm anymore, since the posterior calculations are valid only for Gaussian distributions. Still, in the presence on an uninformative prior, it achieves the following expected pseudo-regret bounds for a general ST-MAB:

Theorem 2.6. *(Gaussian Thompson Sampling regret upper bound on generic ST-MAB [24]) For any $|\mathcal{K}|$ armed stochastic MAB instance Θ, Algorithm 3 achieves the following expected pseudo-regret upper bound:*

$$\mathbb{E}\left[\bar{L}_{T,\Theta}(\mathfrak{U}_{TS})\right] \leq \sum_{i \neq I^*} \left(\frac{18 \ln\left(T\Delta_i^2\right)}{\mu_i} + \frac{25}{\Delta_i}\right) + O(1),$$

where μ_i is the mean of the distribution of arm i and $\Delta_i = \mu_{i^*} - \mu_i$.

However, with some priors, it may not be able to achieve the theoretical bound [33], thus the choice of the prior distribution is very important when

using TS. Despite being one of the oldest heuristic to address the exploration versus exploitation tradeoff, empirical results show that it is highly competitive both on simulated and real-world scenarios [34].

2.3.2 Algorithms for adversarial MAB

In this section, we show how deterministic policies fail to reach the lowest possible order of regret on an adversarial MAB and we present a randomized policy, namely Exponential-weight algorithm for Exploration and Exploitation (Exp3), that achieves it.

2.3.2.1 Deterministic policies applied to Adversarial MABs

The following theorem gives a trivial lower bound on the worst-case regret of any deterministic policy.

Theorem 2.7. *(Worst case regret of any deterministic policy in a A-MAB problem) Any deterministic policy $\mathfrak{U}$, in an A-MAB, has a worst-case regret $L_T^*(\mathfrak{U})$ of at least:*

$$L_T^*(\mathfrak{U}) \geq T\left(1 - \frac{1}{K}\right)$$

Therefore, the weak regret of a deterministic policy in an A-MAB is $O(T)$. As a proof, given the deterministic policy $\mathfrak{U}$, the adversary can choose $r_{i_t,t} = 0$ for all t and $r_{i,t} = 1$ for each $i \neq i_t$ and achieve the worst-case regret of the theorem. Therefore, the main goal of an A-MAB policy is to obtain a sub-linear regret over all the possible adversarial reward assignments.

2.3.2.2 Exp3

A policy that achieves the lowest possible order of regret for a A-MAB problem is Exp3 [9]. Exp3 is a variation of the Softmax algorithm in which the probability $p_{i,t}$ of choosing arm i at round t has the following upper bound:

$$p_{i,t} = (1-\beta)\frac{w_{i,t}}{\sum_{j=1}^{N} w_{j,t}} + \frac{\beta}{K},$$

where β is a parameter that represents the amount of exploration of the algorithm, and $w_{i,t}$ is a weight associated to arm i at round t. This weight gets updated as follows:

$$w_{i,t+1} = \begin{cases} w_{i,t}e^{\eta\frac{r_{i,t}}{\pi_{i,t}}}, & \text{if arm } i \text{ has been pulled at round } t \\ w_{i,t}, & \text{otherwise} \end{cases}$$

The pseudocode of Exp3 is reported in Algorithm 4. An upper bound on the regret of Exp3 is given by the following theorem:

Theorem 2.8. *(Exp3 regret upper bound [9]) At time T, the expected total regret of the Exp3 algorithm applied to an A-MAB problem with $\beta = \min\left(1, \sqrt{\frac{K\log(K)}{(e-1)T}}\right)$ and $\eta = \frac{\beta}{K}$ is:*

$$\mathbb{E}[L_T(\mathfrak{U}_{Exp3})] \leq O\left(\sqrt{TK\log(K)}\right).$$

Algorithm 4 Exp3 algorithm

Require:

β: exploration parameter

η: exponential update parameter

1: Initialize $w_{i,1} = 1$ for each $i \in \mathcal{K}$
2: **for** t=1,2,...,T **do**
3: Choose arm i_t by sampling with probability $p_{i,t}$
4: Play arm i_t
5: Observe reward and update $w_{i,t}$ and $p_{i,t}$
6: **end for**

2.4 Change Detection Test procedures

As introduced in Chapter 1, non-stationarities in MAB problems are often tackled by coupling stationary MAB policies with online procedures that monitor the received samples looking for distribution changes. The parameters that are generally monitored by a change detection procedure are the mean, variance, correlation or spectral density of the process. Some applications of such techniques [7] are manufacturing for quality control, intrusion detection, spam filtering and medical diagnostics.

It is useful to define the following elements regarding changes and Changepoint Detection Test (CDT) in a ANS-MAB:

Definition 8. (Breakpoint) A **breakpoint** is a round t in which at least one arm changes its distribution, i.e., $\exists i \in \mathcal{K}$ s.t. $\mu_{t-1}(i) \neq \mu_t(i)$. From now on ew use the term **changepoint** as synonim for breakpoint.

Definition 9. (Detection point) A **detection point** is a round in which a CDT signals an alarm, that may be a true or a false alarm.

Policies that use these procedures are introduced in Chapter 4. We give now two examples of CDT procedures: the Cumulative Sum control chart

(CUSUM) algorithm, used in Cumulative Sum control chart UCB (CUSUM-UCB) in Section 4.4 and in our novel algorithm CUH in Chapter 5, and the Page-Hinkley Test (PHT) algorithm, used in Adaptive Exploration versus Exploration (Adapt-EvE) in Section 4.3.

2.4.1 CUSUM

The CUSUM procedure keeps track online of the log-likelihood ratio between two hypotheses:

- H_1: the stochastic process generating the samples has a breakpoint at round k.
- H_0: there is no breakpoint in the analysed samples.

The derivation of the algorithm is the following:

$$l_T = \ln\left(\frac{L_{H_1,T}}{L_{H_0,T}}\right) = \ln\left(\frac{\prod_{i=1}^{k-1} p(\theta_0|X_i)\prod_{i=k}^{T} p(\theta_1|X_i)}{\prod_{i=1}^{T} p(\theta_0|X_i)}\right)$$
$$= \ln\prod_{i=k}^{T}\left(\frac{p(\theta_1|X_i)}{p(\theta_0|X_i)}\right) = \sum_{i=k}^{T}\underbrace{\left(\ln\frac{p(\theta_1|X_i)}{p(\theta_0|X_i)}\right)}_{s(i)\text{: CUSUM step}}.$$

We define

$$\underbrace{S(i)}_{\text{CUSUM walk}} = \sum_{j=1}^{i} s(j)$$

and we obtain that the log-likelihood can be computed as follows:

$$l_T = \underbrace{S(T)}_{\text{last CUSUM walk value}} - \underbrace{S(k-1)}_{\text{CUSUM walk value before timestep } k}$$

$$max(l_T) = S(T) - \min_k S(k-1)$$

$$\hat{k} = \arg\min_k S(k-1).$$

Since $\hat{k}$ is the timestep that provides maximum likelihood ratio between the two hypotheses, it is the most probable timestep to contain a breakpoint with respect to the other timesteps. To decide if $\hat{k}$ is a changepoint or not, its log-likelihood ratio must be compared with a threshold h to be tuned according to the specific problem, i.e., $max(l_T) \geq h$ raises an alarm. High h implies fewer false alarms occur at the cost of a higher delay in detecting a changepoint. All the change detection procedures face this tradeoff, thus their average run length (ARL) is analysed, i.e., the average number of timesteps elapsed when an alarm is triggered. In particular, we call ARL_0

the average run length when samples keep coming from the distribution with parameter θ_0, and we call ARL_1 the average run length when samples come from the distribution with parameter θ_1. The quantities of interest for the policies developed for NS-MAB problems are the following:

- $\mathbb{E}[D] = ARL_1$: the mean delay, i.e., the expected number of timesteps required by the CDT to find a change in the monitored distribution.
- $\mathbb{E}[F] = \frac{T}{ARL_0}$: the expected number of false alarms up to time T.

The next step in the CUSUM derivation is to assume a specific probability distribution over the samples, so that the generic probabilities can be substituted with the probability distribution function of such distribution. We give two examples: Bernoulli distributions and Gaussian distributions.

If we assume the distributions to be Bernoulli with means p_0 and p_1 respectively before and after the breakpoint, we obtain the following formula for the CUSUM step:

$$s(i) = \ln\left(\frac{p(\theta_1|X_i)}{p(\theta_0|X_i)}\right) = \ln\left(\frac{p_1^{X_i}(1-p_1)^{1-X_i}}{p_0^{X_i}(1-p_0)^{1-X_i}}\right) = X_i \ln\left(\frac{p_1}{p_0}\right) + (1-X_i)\ln\left(\frac{1-p_1}{1-p_0}\right)$$

If we assume the distributions to be gaussians with means μ_0 and μ_1 and same variance σ^2, we obtain:

$$\begin{aligned} s(i) &= \ln\left(\frac{p(\theta_1|X_i)}{p(\theta_0|X_i)}\right) = \ln\left(\frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{X_i-\mu_1}{\sigma}\right)^2}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{X_i-\mu_0}{\sigma}\right)^2}}\right) = \ln\left(e^{-\frac{1}{2}\frac{(X_i-\mu_1)^2-(X_i-\mu_0)^2}{\sigma^2}}\right) \\ &= -\frac{1}{2\sigma^2}((X_i-\mu_1+X_i-\mu_0)(X_i-\mu_1-X_i+\mu_1)) \\ &= -\frac{1}{2\sigma^2}(\mu_0-\mu_1)\left(X_i - \frac{\mu_0+\mu_1}{2}\right) \end{aligned}$$

Example 4. An example of a CUSUM run can be found in Figure 2.6, where a graphical illustration of the detection delay is present. The samples from the first and second distribution are respectively from $\mathcal{N}(1,1)$ and $\mathcal{N}(2,1)$. We set the CUSUM threshold $h = 10$. The blue line represents the CUSUM walk (i.e., the value of the log-likelihood ratio between the two hypothesis at each timestep), the red line is the minimum value of the walk, and the green line is the threshold to surpass in order to signal an alarm, whose height is the sum of the minimum value of the walk and h. In this example, the walk surpasses the threshold and thus the procedure detects a change at round 60, while the true breakpoint is at round 50. Thus, the delay is of 10 rounds.

In Section 4.4 we develop a modification of the CUSUM algorithm for the bandit setting. Other CDT techniques can be found in [35].

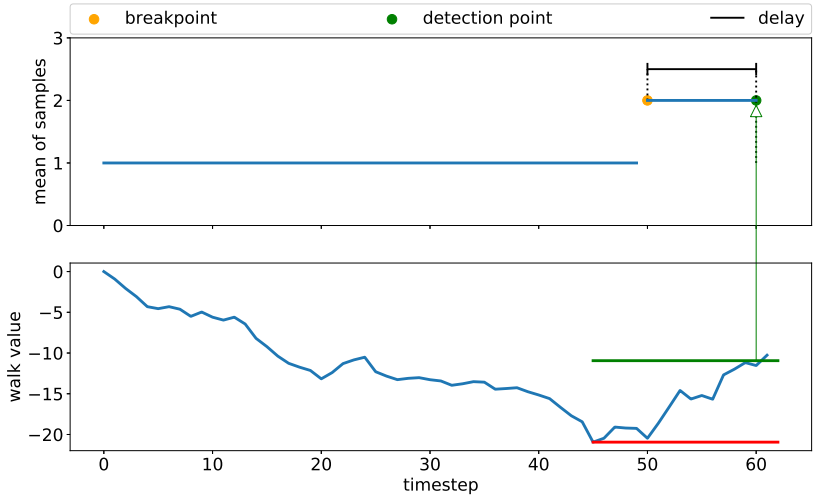


Figure 2.6: Example of a CUSUM run.

2.4.2 PHT

The PHT is very similar to the CUSUM procedure shown in Section 2.4.1, except that it substitutes, in the step formula, the mean of the first M samples with the mean of all the received samples. Let $\bar{X}_s$ be the mean of all the received samples up to timestep s:

$$\bar{X}_s = \frac{1}{s}\sum_{k=1}^{s} X_k.$$

Then, assuming gaussian samples, the value of the PHT walk at timestep t is the sum of all the PHT steps up to timestep t:

$$\underbrace{m_t}_{\text{PHT walk}} = \sum_{s=1}^{t} \underbrace{(X_s - \bar{X}_s + \epsilon)}_{\text{PHT step}}.$$

Finally, like for CUSUM, we compute the timestep with maximum log-likelihood, and compare it with a threshold h:

$$M_t = max\{m_s, s = 1...t\}$$

$$PH_t = M_t - m_t > h \rightarrow \text{ signal alarm}$$

This PHT formulation detects only decreasing changes in a distribution and it is the only one used by the Adapt-EvE policy. As a consequence, its sub-algorithm, which is a stationary policy, will pull each non-optimal arm less and less times, not being able to rapidly detect breakpoints on them. Some examples of PHT runs with different parameters settings are shown in Figure 2.7.

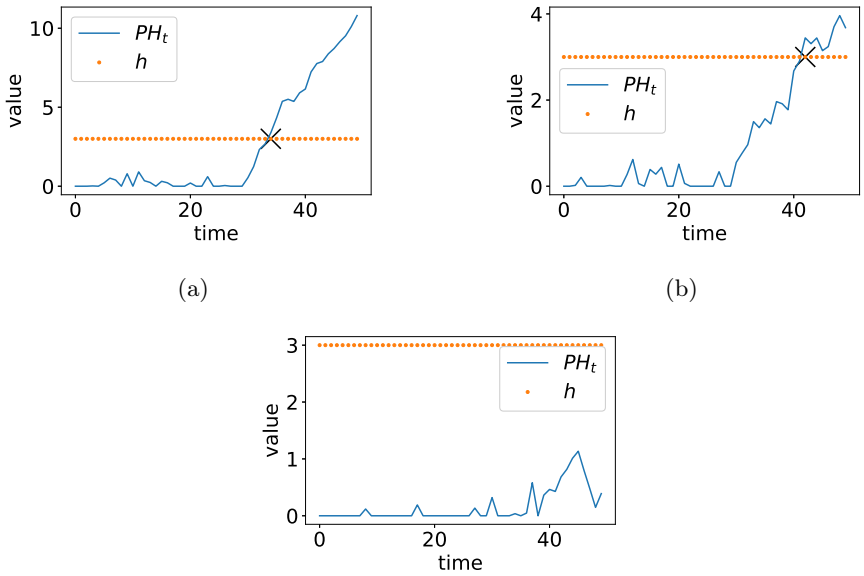


Figure 2.7: Two simulations of the PHT with gaussian data. The first 30 samples are from $\mathcal{N}(1, 0.6)$, then samples are from $\mathcal{N}(0.3, 0.4)$. In (a) the PHT is run with $\epsilon = 0.1$ and $h = 3$, in (b) with $\epsilon = 0.4$ and $h = 3$ and in (c) with $\epsilon = 0.7$ and $h = 3$. The black cross denotes an alarm raised. A small ϵ makes detection faster at the cost of higher probability of false alarms. An ϵ which is too big may also make the procedure fail to recognize a change.

2.4.3 Breakpoint prediction

Since our novel algorithm presented and analysed in Chapter 5.1 uses breakpoint prediction, it is useful to make here a brief introduction on the subject.

Definition 10. (Breakpoint prediction) A **breakpoint prediction** (or **changepoint prediction**) is the estimated round of a breakpoint, made by the CDT.

Definition 11. (Overshoot) We call **overshoot** the difference between the breakpoint prediction and the breakpoint, whenever the breakpoint prediction predicts over the breakpoint and thus suggests samples from the previous distribution. In such case, we say that the CDT has **overshot**.

It is very easy to perform breakpoint prediction with the CUSUM procedure, since it is done automatically online by tracking the timestep k with highest log-likelihood ratio, as shown in Section 2.4.1.

Example 5. An example of a CUSUM run with breakpoint prediction can be found in figure 2.8, where the concepts of breakpoint, detection point, predicted breakpoint and overshoot have a graphical representation. The samples from the first and second distribution are respectively from $\mathcal{N}(1, 1)$ and $\mathcal{N}(2, 1)$. We set the CUSUM threshold $h = 10$. The blue line represents the CUSUM walk, the red line is the minimum value of the walk, and the green line is the threshold to surpass in order to signal an alarm, whose height is the sum of the minimum value of the walk and h. In this example, the walk surpasses the threshold and thus the procedure detects a change at timestep 60, predicts the breakpoint at 45 while the true breakpoint is at timestep 50, thus it overshoots it by 5.

Example 6. Simulations of the CUSUM algorithm with focus on the estimation of the breakpoint k are shown in Figure 2.9, highlighting how both the average predicted breakpoint and the probability of overshoot depend on the detection point. The samples from the first and second distribution are respectively from $\mathcal{N}(1, 0.7)$ and $\mathcal{N}(2, 0.7)$. We set the CUSUM threshold $h = 8$. From these simulations it can be seen that, in practice, breakpoint prediction made with the CUSUM algorithm has a rather low probability of overshoot and does not make big estimation errors.

Example 7. Simulations of the CUSUM algorithm with focus on the breakpoint estimation in case of a false alarm are shown in Figure 2.10, pointing how the breakpoint prediction does not depend on the detection point. The samples from the first and second distribution are respectively from $\mathcal{N}(1, 1)$ and $\mathcal{N}(2, 1)$. We set the CUSUM threshold $h = 2.5$.

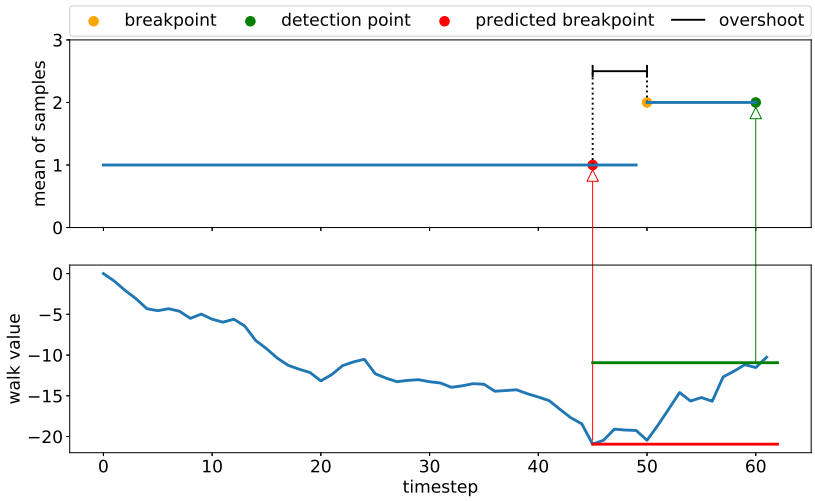
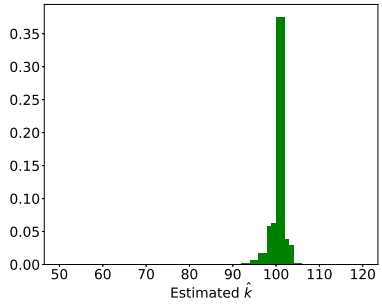
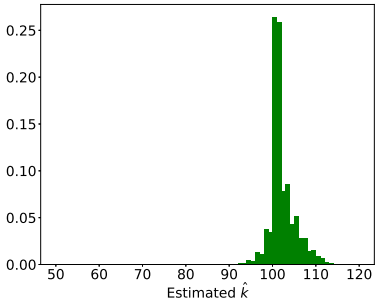


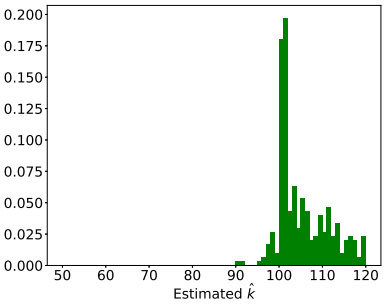
Figure 2.8: Example of a CUSUM run with breakpoint prediction.



(a)



(b)

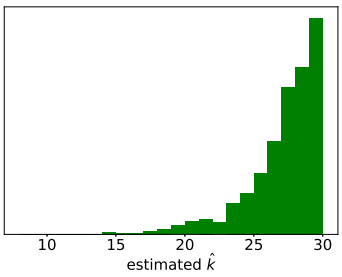


(c)

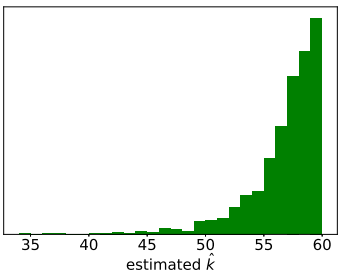
Alarm at	*Avg predicted breakpoint*	*Probability of overshoot*
105	100.12	17.61%
112	101.64	11.15%
120	104.87	7%

(d)

Figure 2.9: Simulations of the CUSUM algorithm with focus on the estimation of the breakpoint k. In (a), (b) and (c) the algorithm detects a change at timesteps respectively 105, 112 and 120. The breakpoint is at timestep 100. The estimated breakpoints and the probability of overshoot for each case are reported in (d).



(a)



(b)

Figure 2.10: Simulations of the CUSUM algorithm, capturing the distribution of the predicted breakpoint $\hat{k}$ when a false alarm is found. In (a) a false alarm is raised at timestep 30, in (b) at timestep 60.

Chapter 3

Stochastic Non-stationary MAB problem formulation

In this chapter, we formalize the stochastic non-stationary MAB problem and we enunciate a lower bound on the pseudo-regret of abruptly changing environments.

3.1 Formulation

A NS-MAB is a stochastic MAB problem in which the distribution of at least one arm changes before T. Formally:

Definition 12. (Non-Stationary MAB) A NS-MAB problem is a tuple $< \mathcal{K}, R >$ where:

- K is the set of possible actions;
- $R := R(r|a,t)$ is an unknown probability distribution of rewards r given the chosen action a at time t, where $\exists a \in \mathcal{K}, t_1, t_2$ s.t. $R(r|a,t_1) \neq R(r|a,t_2)$.

We denote with $\{1,\dots,T\}$ be the decision slots faced by the decision maker, $X_t(i_t)$ be the reward obtained by the decision maker, choosing arm $i_t \in \mathcal{K}$ at timestep t. The rewards $\{X_t(i)\}_{t\geq 1}$ of arm i are modeled as a sequence of independent random variables from potentially different distributions, unknown to the decision maker, and $\mu_t(i) = \mathbb{E}[X_t(i)]$ be the expected reward of arm i at timestep t.

In the case of a NS-MAB, there is not only an optimal arm i^* since the distributions change, but an optimal arm i_t^* is defined for each time step t. We call i_t^* the arm with highest expected reward at timestep t, i.e., $i_t^* = \arg\max_{i\in\mathcal{K}} \mu_t(i)$. Consequently, we define $\Delta_{\mu,T}(i) = min\{\mu_t(i_t^*) - \mu_t(i) : t < T, i \neq i_t^*\}$ as the smallest difference in terms of expected reward, considering all the timesteps $t < T$, between the best arm and arm i.

The regret definition becomes the following:

Definition 13. (Dynamic pseudo-regret) The **dynamic pseudo-regret** of a policy $\mathfrak{U}$ on a NS-MAB problem is defined as:

$$L_T(\mathfrak{U}) = \sum_{t=1}^{T} \mathbb{E}\left(\mu_{i_t^*} - \mu_{i_t}\right)$$

where i_t is the arm chosen by the policy $\mathfrak{U}$ at time t and the expectation is w.r.t. the stochasticity of the policy.

We make the following assumptions:

Assumption 2. Changes are independent from the policy or from the sequence of rewards.

Assumption 2 ensures that the behaviour of the policy does not affect the environment, therefore there exist no decisions that would drive variations on the changes or on the sequence of rewards.

Assumption 3. Only abrupt changes occur.

Assumption 3 is made because this thesis focuses on ANS-MAB.

Assumption 4. Time horizon T and the growth rate of the number of breakpoints Υ_T are known in advance.

Assumption 3 is made because this thesis focuses on fixed-game finite-time analyses. Moreover, since our developed policy uses a CDT, the following two assumptions are commonly made:

Assumption 5. After each breakpoint, there are at least M timesteps in which no other breakpoint occurs.

Assumption 5 ensures that at least for M timesteps after a breakpoint the setting is stationary. This is commonly required by CDT to properly model the null hypotheses.

Assumption 6. For each arm $i \in \mathcal{K}$ and for each $t \leq T-1$, if $\mu_t(i) \neq \mu_{t+1}(i)$, then $|\mu_t(i) - \mu_{t+1}(i)| \geq 2\epsilon$ for a certain $\epsilon > 0$.

Assumption 6 requires that, when the distribution of an arm changes, its mean changes by at least 2ϵ, a quantity big enough to be detected by the CDT. Moreover we assume the arms to be Bernoulli:

Assumption 7. All reward distributions of the arms are Bernoulli distributions, i.e., $R(r|a_i, t) \sim Be(\mu_t(i))$.

Therefore the upper bound of the rewards values is 1 ($B = 1$). Analysing Assumption 5, the empirical average of M Bernoulli samples must be one of the points in $\{0, \frac{1}{M}, \ldots, 1\}$. The minimal non-trivial gap between mean and closest point of arm i is:

$$\lambda(i) = \min_{t \leq T} \left\{ (\mu_t(i) - \epsilon) - \frac{\lfloor (\mu_t(i) - \epsilon)M \rfloor}{M}, \right. \\ \left. \frac{\lceil (\mu_t(i) + \epsilon)M \rceil}{M} - (\mu_t(i) + \epsilon) \right\} \setminus \{0\}$$

We denote the minimal gap of all arms with$\lambda = \min_{i \in \mathcal{K}} \lambda(i)$.

3.2 Lower bound on pseudo-regret of abruptly changing environments

A lower bound on the pseudo-regret of any policy operating in a NS-MAB with abrupt changes is available in the literature.

Theorem 3.1. *(Regret lower bound on an ANS-MAB [10]) Any policy $\mathfrak{U}$ in an ANS-MAB setting have a dynamic expected pseudo-regret s.t.:*

$$\mathbb{E}[L_T(\mathfrak{U})] \geq \sqrt{CT},$$

where $C \in \mathbb{R}^+$ is a constant which depends on the means of the arms of the specific problem taken into consideration and on the Kullback-Leibler divergence between its distributions.

This result must be compared with the bounds previously cited on stationary regret and adversary regret. In the stationary case, a lower bound of $\Omega(\log(T))$ was found for fixed distributions and T allowed to go to infinity. In the adversarial case, a minimax lower bound of $\Omega\left(\sqrt{T}\right)$ was found for finite T. In the bound presented in this section, for each T, the worst case among all possible distributions is considered and the distance between distributions of rewards tend to 0.

Chapter 4

Non-stationary MAB state-of-the-art

In this chapter, we introduce the state-of-the-art policies for the NS-MAB problem. Following the distinction in [6], there are two main approaches to deal with a non-stationary environment:

- *passively adaptive policies*: algorithms that are unaware of the exact instant in which a change happens, which update their decisions based on the most recent observations. Some examples are:
  - Discounted UCB (D-UCB) [36]: the policy couples UCB1 with a moving average, i.e., with a discount factor. More weight is given to more recent samples, since they are better representing the most recent rewards distributions;
  - Sliding Window UCB (SW-UCB) [10]: the policy couples UCB1 with a sliding window, i.e., for a specific arm, only the τ most recent samples are considered in computing its mean reward, where τ is the size of the window;
- *actively adaptive policies*: algorithms that couple a ST-MAB subpolicy with a CDT to monitor the environment for changepoints. We recall that an introduction to what is a CDT has been given in Section 2.4.1. Some examples are:
  - Adapt-EvE [5]: it employs a Page-Hinkley Test [37] as change point detection procedure to detect decreases of mean on the optimal arm and UCB1-Tuned as sub-algorithm. Moreover, whenever a change is detected, a meta-bandit setting is created whose goal is to choose between a reset instance of UCB1-Tuned and a not reset one: in this way, false alarms are inherently managed by the policy. This policy has been developed with a global switch-

Stationary UCB	Non-stationary D-UCB
$\hat{U}_t(i) := \hat{X}_t(i) + \hat{B}_t(i)$	$\hat{U}_t(i,\gamma) := \hat{X}_t(i,\gamma) + \hat{B}_t(i,\gamma)$
$\hat{X}_t(i) = \frac{1}{N_t(i)} \sum_{s=1}^{t} [X_s(i) \mathbb{1}_{i_s=i}]$	$\hat{X}_t(i,\gamma) = \frac{1}{N_t(i,\gamma)} \sum_{s=1}^{t} [\gamma^{t-s} X_{i,s} \mathbb{1}_{i_s=i}]$
$N_t(i) = \sum_{s=1}^{t} [\mathbb{1}_{i_s=i}]$	$N_t(i,\gamma) = \sum_{s=1}^{t} [\gamma^{t-s} \mathbb{1}_{i_s=i}]$
$\hat{B}_t(i) = \sqrt{\frac{2 \ln t}{N_t(i)}}$	$\hat{B}_t(i,\gamma) = 2B \sqrt{\frac{\xi \ln \left(\sum_{j=1}^{K} N_{j,t}(\gamma) \right)}{N_t(i,\gamma)}}$

Table 4.1: UCB1 and D-UCB corresponding quantities.

ing environment as target: indeed the information about all arms stored in the sub-policy is reset when a changepoint is detected;

– CUSUM-UCB [6]: this algorithm couples UCB1 with uniform Exploration (UCB1-Explore) with a CUSUM changepoint detection test for each arm, that looks for both increases and decreases in their means. Whenever a CDT signals a change, the information about its arm stored in UCB1 is reset, thus making this algorithm ideal in per-arm switching environments.

All these policies are solutions to the ANS-MAB problem. In this thesis, we aim at achieving a lower regret with our novel algorithm, presented in Chapter 5. The empirical comparison with these techniques will be performed in Chapter 6.

4.1 D-UCB

D-UCB is an adaptation of the UCB1 algorithm with a discount factor $\gamma \in (0,1)$ that gives more weight to more recent samples, trying to get a better estimate of the instantaneous expected reward of each arm. It has been presented in [36] and analysed in [10].

Let the rewards be bounded in $[0, B]$. As in UCB1, the upper bounds are computed as the sum of the sample mean reward and an exploration bonus, but in D-UCB they both depend on the discount factor γ. The sample mean reward $\hat{X}_t(i,\gamma)$ is the mean of the received samples, where the sample $X_{i,s}$ is discounted by a factor γ^{t-s}. Similarly, in the exploration bonus, both the time elapsed and the number of times an arm has been pulled are discounted. The differences between UCB1 and D-UCB are shown in Table 4.1 and the pseudocode of D-UCB is shown in Algorithm 5.

D-UCB has the following upper bound on the regret.

Theorem 4.1. *(D-UCB regret upper bound [10]) The D-UCB policy, applied to a ANS-MAB problem, choosing* $\gamma = 1 - (4B)^{-1} \sqrt{\Upsilon_T / T}$*, has the following regret upper bound:*

$$\mathbb{E}_\gamma[L_T(\mathfrak{U}_{D-UCB})] = O\left(\sqrt{T \Upsilon_T} \log T \right).$$

Algorithm 5 D-UCB

Require:

γ: discount factor

1: for t from 1 to K, play arm $i_t = t$
2: for t from $K+1$ to T, play arm

$$i_t = \arg\max_{1 \leq i \leq K} \left\{ \hat{U}_t(i, \gamma) \right\}$$

Stationary UCB	Non-stationary SW-UCB
$\hat{U}_t(i) := \hat{Q}_t(i) + \hat{B}_t(i)$	$\hat{U}_t(i,\tau) := \hat{X}_t(i,\tau) + \hat{B}_t(i,\tau)$
$\hat{X}_t(i) = \frac{1}{N_t(i)} \sum_{s=1}^{t} [r_{i,s} \mathbb{1}_{i_s=i}]$	$\hat{X}_t(i,\tau) = \frac{1}{N_t(i,\tau)} \sum_{s=t-\tau+1}^{t} [r_{i,s} \mathbb{1}_{i_s=i}]$
$N_t(i) = \sum_{s=1}^{t} [\mathbb{1}_{i_s=i}]$	$N_t(i,\tau) = \sum_{s=t-\tau+1}^{t} [\mathbb{1}_{i_s=i}]$
$\hat{B}_t(i) = \sqrt{\frac{2 \ln t}{N_t(i)}}$	$\hat{B}_t(i,\tau) = B\sqrt{\frac{\xi \ln(\min\{t,\tau\})}{N_t(i,\tau)}}$

Table 4.2: UCB1 and SW-UCB corresponding quantities.

Algorithm 6 SW-UCB

Require:

τ: sliding window size

1: for t from 1 to K, play arm $i_t = t$
2: for t from $K+1$ to T, play arm

$$i_t = \arg\max_{1 \leq i \leq K} \left\{ \hat{U}_t(i, \tau) \right\}$$

4.2 SW-UCB

SW-UCB is a non-stationary adaptation of UCB1 in which decisions are made taking into account only the $\tau \in \mathbb{N}$ most recent samples, where τ is the size of the sliding window. It has been studied in [10].

Let the rewards be bounded in $[0, B]$. As in UCB1, the upper bounds are computed as the sum of the sample mean reward and an exploration bonus, but in SW-UCB they both depend on the window size τ. The sample mean reward $\hat{X}_t(i,\tau)$ is the mean of the last τ received samples. Similarly, in the exploration bonus, the time elapsed considered is the size of the window and the number of times an arm has been pulled is counted only inside the window. The differences between UCB1 and SW-UCB are shown in Table 4.2. The pseudocode of SW-UCB is shown in Algorithm 6.

SW-UCB has the following upper bound on the regret.

Theorem 4.2. *(SW-UCB regret upper bound [10]) The SW-UCB policy, applied to a ANS-MAB problem, choosing* $\tau = 2B\sqrt{T \ln T / \Upsilon_T}$*, has the following regret upper bound:*

$$\mathbb{E}_\gamma[L_T(\mathfrak{U}_{SW-UCB})] = O\left(\sqrt{T\Upsilon_T \log T}\right).$$

Note that the SW-UCB policy is linear in running time with respect to the number of arms, but is $O(\tau)$ in space since it must store the last τ rewards received.

4.3 Adapt-EvE

Adapt-EvE [5] is an actively adaptive policy that uses UCB1-Tuned as sub-algorithm and employs a PHT [37] to detect decreases in the mean of the optimal arm. Whenever a changepoint is detected, a meta-bandit transient phase starts, whose goal is to choose between two options: reset the sub-algorithm or not. This policy, like other actively adaptive policies, can be generalised with any stationary sub-policy and any CDT in place of the PHT. However, the PHT procedure provably minimizes the expected time before detection [38].

The PHT procedure is explained in Section 2.4.2. When the CDT signals an alarm, the algorithm enters a specific transient Exploration vs Exploitation strategy implemented as a meta-bandit. This transient phase is considered as another bandit problem where the two options are:

- restarting the sub-algorithm from scratch;
- discarding the change detection and keeping the same strategy as before.

In the meta-bandit phase, the policy selects at each round the old bandit or the new bandit and gradually decides if the result of the CDT was a false alarm or not. After M timesteps, the meta-bandit phase finishes and the best meta-option becomes the new core algorithm. This process is shown in the flowchart in Figure 4.1.

Adapt-EvE designers tried experimentally to use past samples after an alarm is raised to bootstrap the sub-policy [5]. They call this technique γ-restart, which is implemented by decreasing the estimation effort (i.e., $N_t(i)$) for all the actions after an alarm is triggered:

$$N_t(i) \to \gamma N_t(i) \quad \forall i \in \{1, \ldots, K\},\ \gamma \in [0, 1]$$

where γ represents the amount of "memory" of the sub-algorithm to be kept. If the alarm is a false alarm, a big γ would help in preserving inertia and

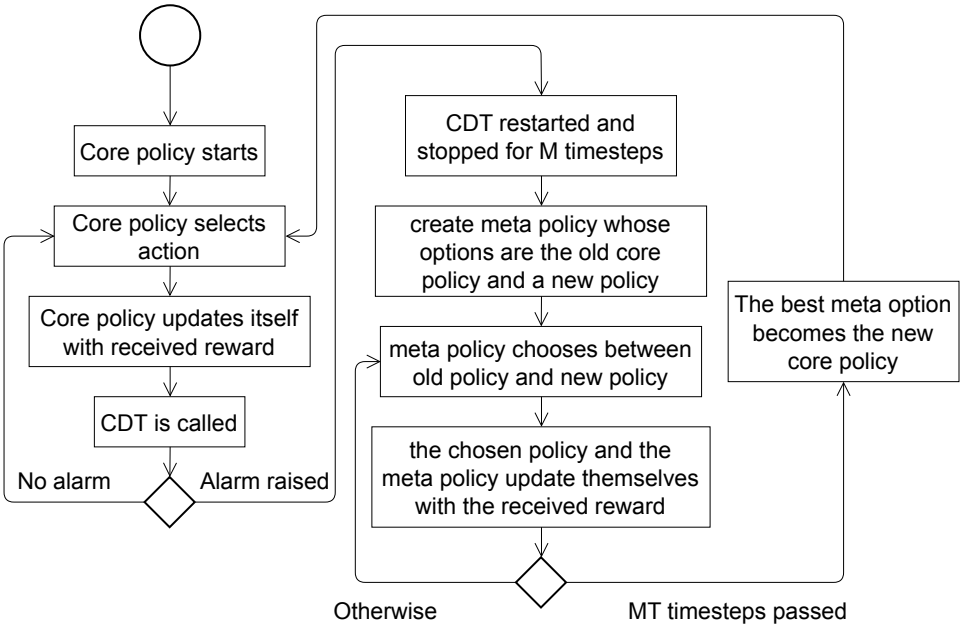


Figure 4.1: Flowchart of the Adapt-EvE algorithm.

keep exploiting. On the other hand, a big γ would hinter the exploration if the CDT was right. Experimentally [5], it turns out that $\gamma = 0$ is the optimal setting, i.e. restarting the sub-algorithm from scratch.

There is no proven upper bound on the regret of Adapt-EvE up to now, thus the parameters of the algorithm must be decided with a trial and error procedure.

4.4 CUSUM-UCB

CUSUM-UCB is an actively adaptive policy which couples UCB1-Explore with an adapted CUSUM for each arm [6], which looks for both increases and decreases in the mean. After UCB1-Explore chooses the arm to pull, the reward returned by the non-stationary environment is fed to both UCB1-Explore and the CDT, whose goal is to monitor the distribution looking for breakpoints. When a CDT signals a change, the information about its arm stored is reset, as shown in figure 4.2. It can be generalised to a policy which employs a generic sub-algorithm in place of UCB1-Explore and a generic CDT in place of CUSUM.

The change detection algorithms have been studied extensively [39]. However, the application of such algorithms usually assumes assumptions which might not hold in the bandit setting. In particular, bandit settings have:

- *Unknown priors*: usually a CDT starts with the knowledge of the prior

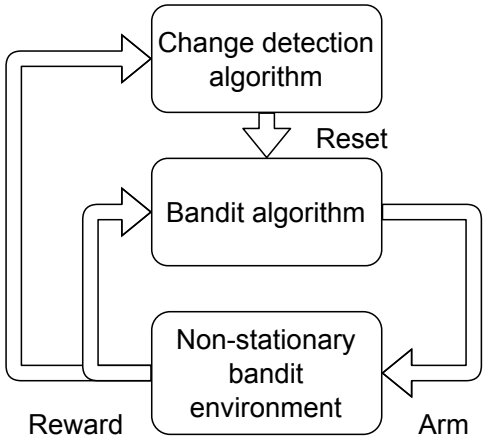


Figure 4.2: Change-detection based framework for ANS-MAB. The problem is tackled by coupling a bandit algorithm with a CDT procedure: at each received reward, the CDT detects a changepoint or not and, as a result, resets the information in the bandit algorithm about that arm or not.

distributions before and after the change. This is not the case in the bandit setting;

- *Insufficient samples*: a CDT needs a certain number of samples to signal a change. This is hard to obtain in a bandit setting, where each arm has its own CDT and the policy tries to exploit the optimal arm only. Thus, the decision maker must intentionally pull each arm to avoid missing changepoints.

The standard CUSUM procedure presented in Section 2.4.1 has such issues and so an adaptation is needed. The first issue is the fact that neither the distribution of samples before the change with mean μ_0, nor the distribution after the change with mean μ_1 are known. The knowledge of μ_0 is substituted with an estimate $\hat{\mu}_0$ made with the first M received samples: during the collection of these M samples, the CDT cannot signal any change since it is acquiring knowledge about the distribution before the change. An estimate of μ_1 is made assuming that $|\mu_1 - \mu_0| = 2\epsilon$. Thus, two independent CUSUM procedures are used, one that looks for increases in the mean, the other that looks for decreases in the mean. As in Section 2.4.1, we give two examples of adapted CUSUM: the former for gaussian samples, the latter for bernoulli ones.

Example 8. We assume now gaussian samples. The first procedure assumes $\mu_1 = \mu_0 + 2\epsilon$ and thus its gaussian update step [6] becomes:

$$s(i) = \frac{2\epsilon}{\sigma^2}(X_i - \hat{\mu_0} - \epsilon) \rightarrow X_i - \hat{\mu_0} - \epsilon,$$

since the $\frac{2\epsilon}{\sigma^2}$ factor can be integrated in the threshold h. The second procedure assumes $\mu_1 = \mu_0 - 2\epsilon$ and thus its update step becomes:

$$s(i) = \frac{2\epsilon}{\sigma^2}(\hat{\mu_0} - X_i - \epsilon) \to \hat{\mu_0} - X_i - \epsilon.$$

Example 9. If otherwise the Bernoulli update step is used, the update steps for increases and decreases become respectively:

$$s(i) = X_i \ln\left(1 + \frac{\epsilon}{\hat{p_0}}\right) + (1 - X_i) \ln\left(1 - \frac{\epsilon}{1 - \hat{p_0}}\right),$$

$$s(i) = X_i \ln\left(1 - \frac{\epsilon}{\hat{p_0}}\right) + (1 - X_i) \ln\left(1 + \frac{\epsilon}{1 - \hat{p_0}}\right).$$

The same threshold h is used for both the procedure detecting increases and the one detecting decreases, making this change detection test symmetric. Once a procedure signals a change, the overall procedure signals such change and both of them are reset. The resulting change detection algorithm is called Two-Sided Cumulative Sum control chart (Two-Sided CUSUM) and detects both increases and decreases in the mean [40]. The CUSUM-UCB policy uses the Gaussian Two-Sided CUSUM. The second issue is solved by adding a uniform exploration over the arms to the sub-algorithm: in this way we have a lower bound on the average of how many times an arm is pulled during an interval. This is the reason why the sub-policy is UCB1-Explore instead of UCB1. The exploration must be tuned according to the specific problem, where more exploration is needed if changes are more frequent and a large number of arms is present.

The UCB1-Explore policy is created by the coupling of the UCB1 policy with random uniform exploration. At each timestep, it chooses a random arm with probability α or it delegates the choice of the arm to UCB1 logic otherwise. Its pseudocode is described in Algorithm 7.

The CUSUM-UCB policy starts by initializing a Two-Sided CUSUM procedure for each arm. Then, at each round, the arm selection is delegated to UCB1-Explore and the received reward is analysed by the CDT procedure: if a change is detected, then both the subpolicy and the CDT procedure of the pulled arm are restarted. The CUSUM-UCB pseudocode is described in Algorithm 8.

Despite being able to estimate the breakpoint, the Two-Sided CUSUM algorithm in the CUSUM-UCB policy is limited to changepoint detection only. In Section 5.1 we will introduce a modification of the CUSUM-UCB policy that takes into account also the breakpoint predictions to improve the performance of the CUSUM-UCB performance.

CUSUM-UCB and Adapt-EvE have some important differences. Adapt-EvE is able to detect only decreases in the mean and, when an alarm is signaled, the sub-policy deletes the information gathered on all the arms.

Algorithm 7 UCB1-EXPLORE

Require:

α: exploration parameter
$\tau(i)$: the timestep, for each arm i, after which data is considered
t: the current timestep

Set $explore = true$ with probability α, set to $false$ otherwise
if *explore* **then**
    $i_t =$ random arm
else
    for $i \in \mathcal{K}$ **do**
        Compute $N_t(i) = \sum_{s=\tau(i)}^{t} \mathbb{1}_{\{I_s=i\}}$
        Compute $X_t(i) = \sum_{s=\tau(i)}^{t} X_s(i)\mathbb{1}_{\{I_s=i\}}$
        Compute $B_t(i) = \sqrt{\frac{\xi \log(t)}{N_t(i)}}$
    end for
    $i_t = \arg\max_i \left(X_t(i) + B_t(i)\right)$
end if
Return arm i_t

Algorithm 8 CUSUM-UCB

Require:

α: exploration parameter
ϵ: CUSUM expected mean difference
h: CUSUM threshold
M: CUSUM initialization parameter

Initialize a CUSUM$(\epsilon, h, M)_i$ CDT for each arm i
Initialize $\tau(i) = 0$ for each arm i
for t up to T **do**
    Make UCB1-EXPLORE(α, τ, t) choose the arm to pull i_t
    Play arm i_t and observe reward $X_t(i_t)$
    if CUSUM$_{i_t}$ $(X_t(i_t)) == 1$ **then**
        Set $\tau(i_t) = t$
        Reset CUSUM$_{i_t}$
    end if
end for

However, if T is small enough (i.e., $T < 10^6$), the policy is still able to react to one increase in the mean of suboptimal arms, as shown in Section 6.2.1. CUSUM-UCB instead detects both increases and decreases in the mean and, when an alarm is detected, the sub-policy deletes the information gathered

only on the changed arm. Moreover, a uniform exploration of suboptimal arms is added to the sub-policy UCB1 so that the CDT is still able to detect changes in the means of suboptimal arms.

The analysis of CUSUM-UCB produces the following upper bound:

Theorem 4.3. *(CUSUM-UCB regret upper bound [6]) The CUSUM-UCB policy, when* $h = \frac{1}{C_1}\log\left(\frac{T}{\gamma_T}\right)$ *and* $\alpha = K\sqrt{\frac{C_2\gamma_T}{C_1 T}\log\left(\frac{T}{\gamma_T}\right)}$*, has an upper bound on the expected non-stationary regret of:*

$$\mathbb{E}\left[L_T(\mathfrak{U}_{CUSUM-UCB})\right] = O\left(\sqrt{T\Upsilon_T \log(\frac{T}{\Upsilon_T})}\right),$$

where

$$C_1 = \min\left(C_1^-, C_1^+\right),$$

$$C_1^- = \log\left(\frac{4\epsilon}{(1-\epsilon)^2}\binom{M}{\lfloor 2\epsilon M\rfloor}(2\epsilon)^M + 1\right),$$

$$C_1^+ = \log\left(\frac{4\epsilon}{(1+\epsilon)^2}\binom{M}{\lceil 2\epsilon M\rceil}(2\epsilon)^M + 1\right),$$

$$C_2 = \log(3) + \frac{2e^{-2\epsilon^2 M}}{\lambda}.$$

Chapter 5

Proposed method

In this chapter, we introduce a novel algorithm, namely CUSUM-UCB with History (CUH), which takes advantage of the breakpoint prediction capability of the CDT procedures. At first we derive an upper bound on the regret of a generic coupling of a MAB and a CDT. Basing on it, we derive an upper bound on the expected pseudo-regret of CUH and we discuss its properties.

5.1 CUSUM-UCB with History: CUH

We propose a new meta-algorithm for the ANS-MAB setting, which combines a ST-MAB sub-policy, namely UCB1-Explore, and a CDT with a breakpoint prediction procedure that must be able to detect both increases and decreases in the mean of a distribution, namely the gaussian Two-Sided CUSUM, previously described in Section 4.4. Thus, our policy is an extension of the CUSUM-UCB policy presented in [6], with the addition of breakpoint prediction. The breakpoint prediction procedure suggests the timestep that maximizes the log-likelihood ratio between the two hypotheses tested by the CUSUM procedure, as explained in Section 2.4.3. The pseudocode of UCB1-Explore can be found in Algorithm 7. The pseudocode of CUH is described in Algorithm 9. At first, a CUSUM procedure is initialised for each arm and each arm is sampled M times, so that each CUSUM computes a precise estimate of each mean. Then, at each round, the arm to pull is chosen by the UCB1-Explore subpolicy and the received reward is fed to the corresponding CDT. If a change is detected, then the subpolicy is partially reset, keeping the information about the samples suggested by the CUSUM procedure.

In Section 5.2 we prove the following upper bound on the expected non-stationary regret of the CUH policy:

Theorem 5.1. *(CUH regret upper bound) The CUH policy, choosing* $W^* = \frac{\xi}{8}\log(T\log(T))$*, applied to a ANS-MAB setting, has the following upper*

bound on the expected pseudo-regret:

$$\mathbb{E}[L_T(\mathfrak{U}_{CUH})] \leq (1 + \mathbb{E}[F] + \gamma_T) \left(A_i + \sqrt{T \log(T)}^{(1-\epsilon)} + 2\zeta\left(\frac{\xi}{2}\right) + T\frac{\alpha}{K} \right) + \\ + \gamma_T \left(\mathbb{E}[D]\frac{K}{\alpha} \right),$$

where

$$A_i = \left\lceil \frac{4\xi \log(T)}{(\Delta_{\mu_T}(i))^2} \right\rceil,$$

$$\mathbb{E}[D] \leq \left(\log(3) + \frac{2}{\lambda} e^{-2\epsilon^2, M} \right)(h+1)$$

$$\mathbb{E}[F] \leq \frac{2T}{(1 - 2e^{-2\epsilon^2 M})e^{Ch}},$$

and $\zeta(n)$ is the Riemann zeta function.

From the theorem, we obtain the following corollary minimizing the upper bound with respect to α and h.

Corollary 5.1. (CUH minimized regret upper bound) Choosing $h = \frac{1}{C_1} \log\left(\frac{T}{\gamma_T}\right)$ and $\alpha = K\sqrt{\frac{C_2 \gamma_T}{C_1 T} \log\left(\frac{T}{\gamma_T}\right)}$, the upper bound on the expected regret of the CUH becomes:

$$\mathbb{E}[L_T(\mathfrak{U}_{CUH})] = O\left(\gamma_T \left(A_i + \sqrt{T \log(T)}^{1-\epsilon} + \sqrt{\frac{C_2}{C_1} T \gamma_T \log(\frac{T}{\gamma_T})} \right) + \right. \\ \left. + \gamma_T \sqrt{\frac{C_1 T}{C_2 \gamma_T} \log(\frac{T}{\gamma_T})} \right) = O\left(\sqrt{T \gamma_T \log\left(\frac{T}{\gamma_T}\right)} \right),$$

where

$$C_1 = \min\left(C_1^-, C_1^+\right),$$

$$C_1^- = \log\left(\frac{4\epsilon}{(1-\epsilon)^2} \binom{M}{\lfloor 2\epsilon M \rfloor} (2\epsilon)^M + 1 \right),$$

$$C_1^+ = \log\left(\frac{4\epsilon}{(1+\epsilon)^2} \binom{M}{\lceil 2\epsilon M \rceil} (2\epsilon)^M + 1 \right),$$

$$C_2 = \log(3) + \frac{2e^{-2\epsilon^2 M}}{\lambda}.$$

This is the first result, up to our knowledge, concerning a MAB policy using breakpoint prediction, and states that the CUH policy achieves a regret whose upper bound has the same order than the one of the CUSUM-UCB policy, despite the presence of bias. The experiments in Chapter 6 show that, in practice, this bias is small and our policy outperforms the state of the art on most scenarios. This result is further discussed in Section 5.3.

Algorithm 9 CUSUM-UCB with History

Require:
T: time horizon
α: Exploration parameter
ϵ: CUSUM expected change parameter
M: CUSUM initialization parameter
h: CUSUM threshold
W^*: maximum amount of history to consider

Initialize $cnt(i) = M$ for each $i \in \mathcal{K}$
Initialize $\tau(i) = 0$ for each $i \in \mathcal{K}$
Initialize CUSUM_i with ϵ, M and h for each arm i
for t in $1, ..., T$ **do**
    if $\exists cnt(i) > 0$ **then**
        $i_t = i$; $cnt(i) = cnt(i) - 1$
    else
        Choose i_t with UCB1-EXPLORE(α, τ, t)
    end if
    Play arm i_t and observe reward $X_t(i_t)$
    $alarm, change_estimate = \text{CUSUM}_{i_t}(X_t(i_t))$
    if $alarm$ **then**
        $\tau(i_t) = \max\{change_estimate, t - W^*\}$
        $cnt(i_t) = M$
        Reset CUSUM_{i_t}
    end if
end for

5.2 Analysis

We structure the proof of the CUH policy upper bound on the pseudo-regret on ANS-MAB settings as follows: in Section 5.2.1 we analyse the proposed meta-algorithm with a generic sub-policy and a generic CDT, without considering breakpoint predictions; then, in Section 5.2.2 we add a specific sub-policy to the analysis, i.e., UCB1-Explore, and we introduce breakpoint prediction to the generic CDT in Section 5.2.3; next, in Section 5.2.4 we consider CUSUM as the specific CDT and we analyse the three possible cases in which breakpoint prediction is used, i.e., true alarms with and without overshoot in Section 5.2.5 and false alarms in Section 5.2.6: all these cases introduce a bias in the bandit which depends on the CDT which is used, and we analyse it for the CUSUM procedure in Section 5.2.7; at this point, all the consequences of having a breakpoint prediction procedure in a policy have been analysed and the meta-algorithm with generic sub-policy and generic CDT with breakpoint prediction is tackled in Section 5.2.8; finally,

in Section 5.2.9 we explain how limiting the breakpoint predictions up to a maximum number of samples before the detection point helps in limiting bias in the estimates made by the policy, and an upper bound on the pseudo-regret is developed in Section 5.2.10.

5.2.1 Analysis of the meta-algorithm without breakpoint prediction

The CUH policy can be generalised to a generic ST-MAB bandit sub-algorithm $\mathfrak{U}$ and a generic CDT which provides also a breakpoint prediction mechanism.

We analyse the policy disregarding the breakpoint prediction procedure, i.e., when a changepoint is detected, no history about the previous arm pulls is used. Consider the scenario in Figure 5.1. We call *phase* the interval of time starting from a changepoint (or from timestep 1) or a false alarm and ending with the next breakpoint (or with timestep T) or a false alarm. Let $L_t(\mathfrak{U}_{SUB})$ be an upper bound on the regret of the stationary sub-policy $\mathfrak{U}_{SUB}$ on a stationary setting with time horizon t. Then, the regret of the full policy $\mathfrak{U}$ obtained by coupling the sub-policy $\mathfrak{U}_{SUB}$ with the CDT is upper bounded by:

$$\begin{aligned} L_T(\mathfrak{U}) &\leq L_{\phi_1}(\mathfrak{U}_{SUB}) + \sum_{i=1}^{F} L_{\phi_{f_i}}(\mathfrak{U}_{SUB}) + \sum_{i=1}^{\gamma_T} D_i + \sum_{i=1}^{\gamma_T} L_{\phi_{c_i}}(\mathfrak{U}_{SUB}) \\ &\leq L_T(\mathfrak{U}_{SUB}) + \sum_{i=1}^{F} L_T(\mathfrak{U}_{SUB}) + \sum_{i=1}^{\gamma_T} D_i + \sum_{i=1}^{\gamma_T} L_T(\mathfrak{U}_{SUB}), \end{aligned}$$

where ϕ_1 is the duration of the first phase, i.e., the one starting at timestep 1 and ending at the first breakpoint, F is the number of false alarms, ϕ_{f_i} is the duration of the i-th phase after a false changepoint, ϕ_{c_i} is the duration of the i-th phase after a true changepoint and D_i is the delay employed by the CDT in detecting the i-th true changepoint, using the samples chosen by the bandit policy $\mathfrak{U}_{SUB}$. Taking the expected value over the experiments and of the setting stochasticity, the upper bound becomes:

$$\mathbb{E}[L_T(\mathfrak{U})] \leq (1 + \mathbb{E}[F]_{\mathfrak{U}_{SUB}} + \gamma_T)\mathbb{E}\left[L_T(\mathfrak{U}_{SUB})\right] + \gamma_T \mathbb{E}[D]_{\mathfrak{U}_{SUB}},$$

where $\mathbb{E}[F]_{\mathfrak{U}_{SUB}}$ and $\mathbb{E}[D]_{\mathfrak{U}_{SUB}}$ are respectively the expected number of false alarms and the expected detection delay of the CDT, when fed only with samples chosen by the bandit policy $\mathfrak{U}_{SUB}$, and, thus, they are different from $\mathbb{E}[F]$ and $\mathbb{E}[D]$, which are independent from the policy and assume the CDT receives a sample at each timestep. Since $\mathfrak{U}_{SUB}$ is bandit algorithm for the stationary setting, it chooses what it believes it is the optimal arm more and more as time passes, pulling sub-optimal arms less and less. As only an arm is pulled at each timestep, we can deduce the following inequalities:

$$\mathbb{E}[F]_{\mathfrak{U}_{SUB}} \leq \mathbb{E}[F],$$

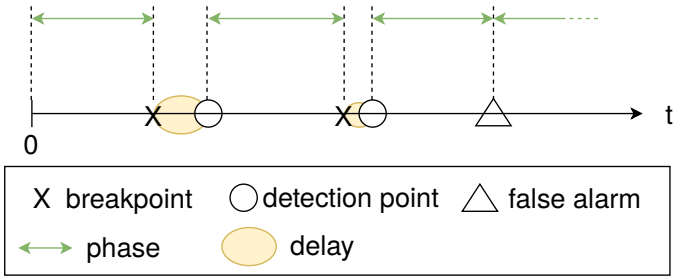


Figure 5.1: Example of non-stationary setting and a CDT that monitors each arm at each timestep. The CDT is not able to instantly detect the breakpoints and thus signals detection points with some delay. The interval of time inside each phase can be considered as a stationary bandit.

$$\mathbb{E}[D]_{\mathfrak{U}_{SUB}} \geq \mathbb{E}[D].$$

Remark that this analysis is independent from the sub-policy $\mathfrak{U}_{SUB}$ and from the CDT employed. The stationary policy $\mathfrak{U}_{SUB}$ may take a very large amount of rounds to pull a sub-optimal arm $\mathbb{E}[D]$ times and detect a changepoint on it. As a result, the sub-policy must explore, in some way, sub-optimal arms to detect changes on them. Suppose the sub-policy $\mathfrak{U}_{SUB}$ chooses a random arm uniformly with probability α. Then, at least $\mathbb{E}[D]$ observations from a generic arm i are obtained on average after $\mathbb{E}[D]\frac{K}{\alpha}$ timesteps, and we can make the following upper bound:

$$\mathbb{E}[D]_{\mathfrak{U}_{SUB}} \leq \mathbb{E}[D]\frac{K}{\alpha}.$$

The upper bound on the pseudo-regret of the policy $\mathfrak{U}$ becomes:

$$\mathbb{E}[L_T(\mathfrak{U})] \leq (1 + \mathbb{E}[F] + \gamma_T)\mathbb{E}[L_T(\mathfrak{U}_{SUB})] + \gamma_T \left(\mathbb{E}[D]\frac{K}{\alpha}\right).$$

5.2.2 UCB1-EXPLORE with a generic CDT

We take a further step towards the analysis of our algorithm CUH substituting the generic sub-policy with UCB1-Explore coupled with a generic CDT, characterised by $\mathbb{E}[D]$ and $\mathbb{E}[F]$ which are assumed to be known. Following the proof line present in [6], we can prove that:

Theorem 5.2. *(Upper bound of regret of UCB1-Explore coupled with a generic CDT) Using the UCB1-Explore policy coupled with a generic CDT on a non-stationary MAB problem, its expected pseudo-regret satisfies:*

$$\mathbb{E}[\tilde{N}_T(i)] \leq (\gamma_T + \mathbb{E}[F] + 1)(A_i + 2\zeta(2\xi)) + \gamma_T \left(\mathbb{E}[D]\frac{K}{\alpha}\right) + T\frac{\alpha}{K},$$

where:

$$A_i = \left\lceil \frac{4\xi \log(T)}{(\Delta_{\mu_T}(i))^2} \right\rceil$$

and ζ is the zeta Riemann function.

The proof proceeds as follows. Let:

- i_t: the arm selected by our algorithm at time t;
- i_t^*: the optimal arm at time t.

The expected number of times $\mathbb{E}[\tilde{N}_T(i)]$ an arm i is pulled up to time T, but it was not optimal in such instants can be decomposed into two terms: one due to the uniform exploration and another due to the selection rule given by the UCB1 algorithm. When choosing which arm to pull, with probability α the policy pulls a random arm, otherwise it selects the arm based on the upper confidence bound on the mean of the arms. Since the probability of selecting at random the arm i is $\frac{1}{K}$ in the uniform exploration case, we have that:

$$\mathbb{E}[\tilde{N}_T(i)] \leq \sum_{t=1}^{T} \left(\alpha \frac{1}{K} + (1-\alpha) P_{\{i_t = i \neq i_t^*\}} \right)$$

$$\leq T \frac{\alpha}{K} + \underbrace{\sum_{t=1}^{T} P_{\{i_t = i \neq i_t^*\}}}_{\mathbb{E}[Q_T(i)]} .$$

where $P_{\{i_t = i \neq i_t^*\}}$ is the probability that the UCB1 logic chooses a suboptimal arm at round t.

Let us focus now on the second term $\mathbb{E}[Q_T(i)] = \sum_{t=1}^{T} P_{\{i_t = i \neq i_t^*\}}$, which does not take into account the random arm exploration anymore. Recalling that the policy resets each time a change is detected, we consider two cases:

- when the monitored distribution did not change. Let $\tau_t(i)$ be the timestep of the last detection point before t. We denote with $\mathcal{T}_i$ the set:

  $$\mathcal{T}_i := \{t \in \{1, \ldots, T\} : \mu_s(i) = \mu_t(i) \text{ and } \tau_t(i) < s \leq t,\, t \geq \tau_t(i) + A_i\};$$

- when the monitored distribution has changed but the policy has not detected the change yet. All these intervals are exactly the delay of the CDT in detecting the changes and we consider these pulls as all wrong pulls. However, as explained in Section 5.2.1, if the change happens on a sub-optimal arm, the policy may take a very large number of

timesteps to collect the observations required for detecting the change. Being $\mathbb{E}[D]$ the average number of observations required for detecting a change on an arm, these observations can be obtained by letting the policy pull a random arm with probability α, so that, on average, at least $\mathbb{E}[D]$ samples are collected after $\mathbb{E}[D]\frac{K}{\alpha}$ timesteps.

So, we write:

$$\mathbb{E}[Q_T(i)] \leq \gamma_T \left(\mathbb{E}[D] \frac{K}{\alpha} \right) + \sum_{t \in \mathcal{T}_i} P_{\{i_t = i \neq i_t^*\}}.$$

The event $\{i_t = i \neq i_t^*\}$ occurs if $\{\bar{X}_t(i) + C_t(i) \geq \bar{X}_t(i_t^*) + C_t(i_t^*)\}$. We want to upper bound the probability of the latter one. It is true that:

$$\begin{aligned}
&\underbrace{\{\bar{X}_t(i) + C_t(i) \geq \bar{X}_t(i_t^*) + C_t(i_t^*)\}}_{D} \subseteq \\
&\underbrace{\{\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\}}_{A} \cup \underbrace{\{\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*)\}}_{B} \cup \\
&\underbrace{\{\mu_t(i_t^*) - \mu_t(i) \leq 2C_t(i)\}}_{C},
\end{aligned}$$

Using simple Boolean algebra, we obtain:

$$\begin{aligned}
D \implies A \vee B \vee C &\equiv \neg D \vee \neg(\neg A \wedge \neg B \wedge \neg C) \\
&\equiv \neg(\neg A \wedge \neg B \wedge \neg C) \vee \neg D \equiv (\neg A \wedge \neg B \wedge \neg C) \implies \neg D.
\end{aligned}$$

So, we want to upper bound the probability of the following set of events:

$$\begin{cases}
\underbrace{\mu_t(i) + C_t(i) > \bar{X}_t(i)}_{\neg A} \\
\underbrace{\bar{X}_t(i_t^*) > \mu_t(i_t^*) - C_t(i_t^*)}_{\neg B} \\
\underbrace{\mu_t(i_t^*) - \mu_t(i) > 2C_t(i)}_{\neg C}
\end{cases},$$

which implies $\underbrace{\bar{X}_t(i) + C_t(i) < \bar{X}_t(i_t^*) + C_t(i_t^*)}_{\neg D}$. Indeed, summing the three inequalities, we obtain:

$$\begin{aligned}
&\mu_t(i) + C_t(i) + \bar{X}_t(i_t^*) + \mu_t(i_t^*) - \mu_t(i) > \bar{X}_t(i) + \mu_t(i_t^*) - C_t(i_t^*) + 2C_t(i) \\
&\bar{X}_t(i) + C_t(i) < \bar{X}_t(i_t^*) + C_t(i_t^*),
\end{aligned}$$

whose negation is:

$$\bar{X}_t(i) + C_t(i) \geq \bar{X}_t(i_t^*) + C_t(i_t^*).$$

Summing up the previous results, we obtain:

$$\begin{aligned}\mathbb{E}[Q_T(i)] &\leq \gamma_T \left(\mathbb{E}[D]\frac{K}{\alpha} \right) + \sum_{t \in \mathcal{T}_i} P_{\{i_t = i \neq i_t^*\}} \\ &\leq \gamma_T \mathbb{E}[D] + \sum_{t \in \mathcal{T}_i} \Bigg(\underbrace{P(\bar{X}_t(i) \geq \mu_t(i) + C_t(i))}_{P(A)} + \\ &+ \underbrace{P(\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*))}_{P(B)} + \underbrace{P(\mu_t(i_t^*) - \mu_t(i) \leq 2C_t(i))}_{P(C)} \Bigg)\end{aligned}$$

The truthfulness of the event C depends on $C_t(i)$, which depends on $N_t(i)$. In particular, when $N_t(i)$ is high enough, the event C has zero probability. We know that $(\Delta_{\mu_T}(i))^2 \leq \mu_t(i_t^*) - \mu_t(i)$ by definition. Thus it must be that:

$$\begin{aligned}\Delta_{\mu_T(i)} &\leq \mu_t(i_t^*) - \mu_t(i) \\ &\leq 2C_t(i) \\ &= 2\sqrt{\frac{\xi \ln(N_t)}{N_t(i)}} \\ &\leq 2\sqrt{\frac{\xi \ln(T)}{N_t(i)}}\end{aligned}$$

and so:

$$(\Delta_{\mu_T}(i))^2 \leq \frac{4\xi \ln(T)}{N_t(i)},$$

which brings to the following condition for the event C to happen:

$$N_t(i) \leq \frac{4\xi \ln(T)}{(\Delta_{\mu_T}(i))^2}.$$

Let us denote from now on the minimum number of pulls for which the bounds are tight enough with:

$$A_i := \left\lceil \frac{4\xi \log(T)}{(\Delta_{\mu_T}(i))^2} \right\rceil.$$

The event $\{i_t = i \neq i_t^*, N_t(i) < A_i\}$ occurs at the beginning of the run and right after the CDT is reset, since $N_t(i)$ is the number of times the algorithm pulls arm a_i up to time t from the last reset of the CDT. This occurs $\gamma_T + F_i + 1$ times, i.e., once each time a true change is detected, once for each false positive change detection and once at the beginning of the execution of the policy. Since when $N_t(i) < A_i$ the bounds are not tight

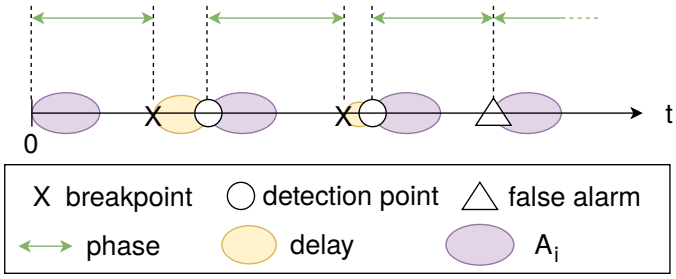


Figure 5.2: Example of an execution of UCB1-Explore with a generic CDT in a ANS-MAB setting. Each phase can be seen as a separate ST-MAB setting. At the beginning of each phase, UCB1-Explore spends A_i timesteps.

enough, we consider all these pulls wrong. Defining $\tilde{\mathcal{T}}_i := \{t \in \mathcal{T}_i | N_t(i) \geq A_i\}$, we have:

$$\mathbb{E}[Q_T(i)] \leq (\gamma_T + F_i + 1)A_i + \gamma_T \left(\mathbb{E}[D]\frac{K}{\alpha}\right) + \sum_{t \in \tilde{\mathcal{T}}_i} \left(\underbrace{P(\bar{X}_t(i) \geq \mu_t(i) + C_t(i))}_{P(A)} + \underbrace{P(\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*))}_{P(B)} \right).$$

The current situation is represented in Figure 5.2. In this specific example, there are two change points and three detection points: the last detection point is a false alarm. All the timesteps that are not in colored regions are the ones considered in $\tilde{\mathcal{T}}_i$.

Using the Chernoff-Hoeffding inequality [28], we obtain:

$$\begin{aligned} P(A) = P\left(\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\right) &\leq e^{-2N_t(i)C_t(i)^2} \\ &= e^{-2N_t(i)\frac{\xi \log(N_t)}{N_t(i)}} \\ &\leq N_t^{-2\xi} \end{aligned}$$

$$P(B) = P\left(\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) + C_t(i_t^*)\right) \leq N_t^{-2\xi}.$$

Hence:

$$\begin{aligned} &\sum_{t \in \tilde{\mathcal{T}}_i} (P(A) + P(B)) \leq \sum_{t \in \tilde{\mathcal{T}}_i} 2N_t^{-2\xi} \leq 2 \sum_{m=0}^{\gamma_T + F_i} \sum_{s=1}^{l_m} s^{-2\xi} \\ &\leq 2 \sum_{m=1}^{\gamma_T + F_i + 1} \sum_{s=1}^{\infty} s^{-2\xi} = 2 \sum_{m=1}^{\gamma_T + F_i + 1} \zeta(2\xi) = (\gamma_T + F_i + 1)2\zeta(2\xi), \end{aligned}$$

where l_m is the length of the m-th run of the algorithm between resets and $\zeta(n) = \sum_{i=1}^{\infty} i^{-n}$ is the Riemann zeta function. We remark that the Riemann

zeta function $\zeta(n)$ converges for each $n > 1$ [41], thus, in this case it converges if $\xi > \frac{1}{2}$.

Summing all the bounds, we obtain:

$$\mathbb{E}[Q_T(i)] \leq (\gamma_T + \mathbb{E}[F] + 1)(A_i + 2\zeta(2\xi)) + \gamma_T \left(\mathbb{E}[D] \frac{K}{\alpha} \right).$$

At last, we add back the pulls from the uniform exploration:

$$\mathbb{E}[\tilde{N}_T(i)] \leq (\gamma_T + \mathbb{E}[F] + 1)(A_i + 2\zeta(2\xi)) + \gamma_T \left(\mathbb{E}[D] \frac{K}{\alpha} \right) + T \frac{\alpha}{K}.$$

Note that, since each reward is in $[0, 1]$:

$$\begin{aligned} \mathbb{E}[L_T(\mathfrak{U})] &= \mathbb{E}\left[\sum_{t=1}^{T} (X_t(i_t^*) - X_t(i_t)) \right] \\ &\leq \mathbb{E}\left[\sum_{i \in \mathcal{K}} \tilde{N}_T(i) \right]. \end{aligned}$$

Therefore it is possible to upper-bound $L_T(\mathfrak{U})$ by upper-bounding $\tilde{N}_T(i)$. Then,

$$\mathbb{E}[L_T(\mathfrak{U})] \leq (\gamma_T + \mathbb{E}[F] + 1)(A_i + 2\zeta(2\xi)) + \gamma_T \left(\mathbb{E}[D] \frac{K}{\alpha} \right) + T \frac{\alpha}{K}.$$

Note that a similar result could have been found directly substituting the upper bound on the expected regret of the UCB1-Explore sub-policy $\mathfrak{U}_{SUB}$ on a ST-MAB into the bound found in Section 5.2.1, thus obtaining:

$$\mathbb{E}[L_T(\mathfrak{U})] \leq (1 + \mathbb{E}[F] + \gamma_T) \left(A_i + 2\zeta(2\xi) + T \frac{\alpha}{K} \right) + \gamma_T \left(\mathbb{E}[D] \frac{K}{\alpha} \right)$$

5.2.3 UCB1-EXPLORE with a generic CDT with breakpoint prediction

We consider now the case of UCB1-Explore coupled with a CDT which, after a reset ot an arm a_i, keeps information about a certain amount of the most recent past history of the arm. The amount of history to be considered is given by the breakpoint prediction procedure of the selected CDT. As shown in Section 5.2.2, when no history is used, the situation is the one in Figure 5.1.

When some history is used after a reset, what happens is depicted in Figure 5.3. Some rounds that were previously belonging to the A_i regions now overlap with the timesteps in the regions of the CDT detection delay. Ideally, we always want to use as history all the data belonging to the period of the detection delay before the last detection point. However, the

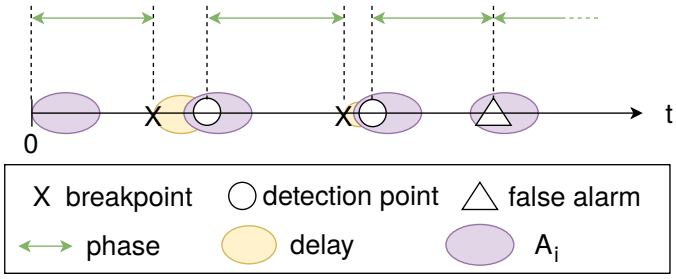


Figure 5.3: Example of a UCB1-Explore run with breakpoint prediction and no overshoot. The effect of breakpoint prediction is to take some of the A_i required samples from the past observations.

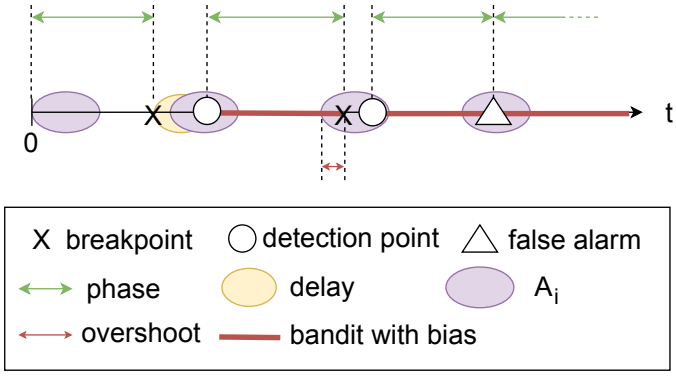


Figure 5.4: Example of a UCB1-Explore run with breakpoint prediction and some overshoot. The effect of breakpoint prediction is to take some of the required A_i samples from past observations, at the cost of some bias in the estimates of UCB1-Explore.

breakpoint prediction procedure may overshoot the changepoint, as shown in Figure 5.4. In such a case, some history from the previous distribution is used for the decision of the next arm to play and therefore the computed estimates becomes biased until the next reset. Moreover, some bias, but in different amounts, is present also in the cases of true alarm without overshooting and false alarm: the past samples suggested by the CDT are not perfectly representative of the original distribution since they are obtained after an alarm has been raised. The optimal scenario, i.e., the scenario with perfect breakpoint prediction, is shown in Figure 5.5. If one is capable of providing a CDT providing always correct breakpoint predictions, we would not have to take into account the presence of bias.

To expand the analysis of the CUH policy taking into account breakpoint prediction, we must consider the following three cases:

- we predict after a true alarm and we do not overshoot the change point: analysed in Section 5.2.5.

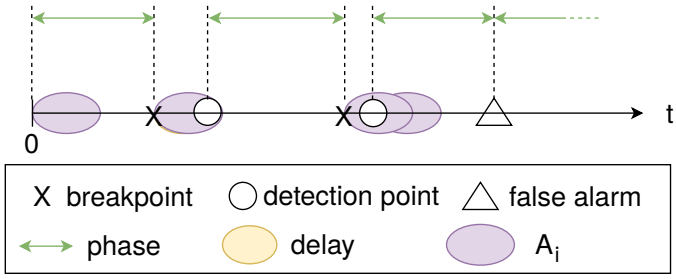


Figure 5.5: Example of a UCB1-Explore run with perfect breakpoint prediction. When a true or a false alarm is raised, samples from the most recent changepoint are kept. In the case of false alarms, the sub-policy is able to start from where it has been interrupted without any side effect.

- we predict after a true alarm and we overshoot the change point: analysed in Section 5.2.5.
- we predict after a false alarm: analysed in Section 5.2.6.

5.2.4 UCB1-EXPLORE with CUSUM

To further advance in the analysis of the CUH policy, we consider the Two-Sided CUSUM procedure in place of the generic CDT. As proven in [6], the expected delay and expected number of false alarms up to time T of the Two-Sided CUSUM algorithm are upper bounded by:

$$\mathbb{E}[D] \leq \left(\log(3) + \frac{2}{\lambda} e^{-2\epsilon^2 M}\right)(h+1),$$

$$\mathbb{E}[F] \leq \frac{2T}{\left(1 - 2e^{-2\epsilon^2 M}\right) e^{C_1 h}},$$

where:

- $C_1 = \min(C_1^-, C_1^+)$;
- $C_1^- = \log\left(\frac{4\epsilon}{(1-\epsilon)^2}\binom{M}{\lfloor 2\epsilon M \rfloor}(2\epsilon)^M + 1\right)$;
- $C_1^+ = \log\left(\frac{4\epsilon}{(1+\epsilon)^2}\binom{M}{\lceil 2\epsilon M \rceil}(2\epsilon)^M + 1\right)$.

Substituting these values in the results obtained in Section 5.2.2, we obtain an upper bound on the pseudo-regret of the CUH policy without breakpoint prediction. In the next sections, we introduce breakpoint prediction in the analysis.

5.2.5 Analysis of breakpoint prediction for true alarms with CUSUM

We now focus on breakpoint prediction with the CUSUM procedure. Let:

- $\tilde{t}_j$ be the timestep of the j-th true breakpoint;
- $\hat{t}_j$ be the estimation of $\tilde{t}_j$ made with the breakpoint prediction procedure of the chosen CDT, after an alarm has been triggered.

At the beginning of each phase, instead of considering A_i wrong pulls necessary to make the event C not possible as in Section 5.2.2, the breakpoint procedure takes into account some samples before the detection point, thus having a larger values of the arms pull $N_t(i)$, which in its turn makes achieving the the collection of A_i samples per arm faster. Let D_i^j be the detection delay of the CDT in detecting the j-th breakpoint on arm i. In the case of breakpoint prediction without overshooting, after the j-th detection point there are no more A_i wrong pulls, but there are:

$$\begin{aligned} A_i \to & \overbrace{\max\left(A_i - D_i^j + \left(\hat{t}_j - \tilde{t}_j\right), 0\right)}^{\text{with breakpoint prediction}} \\ & = \mathbb{E}_{A_i - D_i^j + (\hat{t}_j - \tilde{t}_j) \geq 0}\left[A_i - D_i^j + \left(\hat{t}_j - \tilde{t}_j\right)\right] + \mathbb{E}_{A_i - D_i^j + (\hat{t}_j - \tilde{t}_j) < 0}[0] \\ & = \mathbb{E}_{A_i - D_i^j + (\hat{t}_j - \tilde{t}_j) \geq 0}\left[A_i - D_i^j + \left(\hat{t}_j - \tilde{t}_j\right)\right] \end{aligned}$$

We do not know the exact distribution of the detection delays D_i^j. However, we can derive a lower bound on the history estimated by the CUSUM algorithm analysing how it works. Since the CUSUM algorithm signals a change whenever its walk exceeds its minimum value by h, the estimated change will always be at least $\frac{h}{ms}$ steps behind the current timestep, where:

$$\begin{aligned} ms &= \max_{y_k, \hat{\mu_0}} (y_k - \hat{\mu_0} - \epsilon) \\ &= 1 - \epsilon \end{aligned}$$

is the maximum step the CUSUM statistics performs towards the threshold h in a single timestep. Thus:

$$\max\left(A_i - D_i^j + \left(\hat{t}_j - \tilde{t}_j\right), 0\right) \leq A_i - \frac{h}{1-\epsilon}$$

However, these $\frac{h}{1-\epsilon}$ recovered observations are not fully representative of the past distribution since their mean must be conditioned on the fact that they raised an alarm on the CUSUM procedure. As a result, counterintuitively, they introduce some bias in the estimates made by the UCB1-Explore in the next stationary phase.

Similarly, in the case of breakpoint prediction with overshooting, at least $\frac{h}{1-\epsilon}$ samples are recovered. However, some of them are from the previous distribution, thus making the next bandit phase biased.

5.2.6 Analysis of change instant prediction for false alarms with CUSUM

In the case of a false alarm, the breakpoint prediction procedure returns the most probable change point, which may not be the correct one, i.e., the instant of the previous detection point. Let J_j be the amount of history selected after the j-th false alarm. Similarly to the case with true alarms in Section 5.2.5, after the j-th detection point, there are no more A_i wrong pulls, but there are:

$$A_i \rightarrow \overbrace{\max\left(A_i - J_j, 0\right) \leq A_i - \mathbb{E}_{J \leq A_i}[J]}^{\text{with breakpoint prediction}}$$

The amount of history selected can be lower-bounded by:

$$J \leq \frac{h}{1-\epsilon}.$$

Thus:

$$\max\left(A_i - J_j, 0\right) \leq A_i - \frac{h}{1-\epsilon}.$$

Again, as in the case of true alarm, these $\frac{h}{1-\epsilon}$ samples are not representative of their distribution since their mean must be conditioned on the fact that they raised a false alarm. Thus, counterintuitively, recovering past samples after a false alarm introduces bias in the estimates made by the UCB1-Explore in the next stationary phase.

5.2.7 Bias analysis due to breakpoint prediction with CUSUM

We analyse how UCB1-Explore works after the CUSUM algorithm selects some samples from the previous distribution. Let W be the amount of samples in the history selected by CUSUM. As worst case, we consider these W samples all coming from a different distribution than the one of the future samples. This is equivalent to a bandit problem where the samples come from:

$$\begin{cases} \text{distribution } \theta_0, & \text{if } t \leq W \\ \text{distribution } \theta_1, & \text{otherwise} \end{cases}.$$

We are only interested in the regret for $t \in [W, T]$. We indicate the number of times arm i has been selected in the interval $t \in [W, T]$ but it was not optimal with $\tilde{N}_{W:T}(i)$.

$$\tilde{N}_{W:T}(i) = \sum_{t=W}^{T} \mathbb{1}_{\{i_t = i \neq i_t^*\}} \leq (T-W)\frac{\alpha}{K} + \underbrace{\sum_{t=W}^{T} \mathbb{1}_{\{i_t = i \neq i_t^*\}}}_{Q_{W:T}(i)},$$

where $Q_{W:T}(i)$ does not take into account the random arm exploration anymore. As in Section 5.2.2:

$$Q_{W:T}(i) \leq \sum_{t=W}^{T} \Bigg(\mathbb{1}_{\{\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\}} + \mathbb{1}_{\{\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*)\}} + \mathbb{1}_{\{\mu_t(i_t^*) - \mu_t(i) \leq 2C_t(i)\}} \Bigg).$$

The third event never happens if we impose:

$$2C_t(i) \leq \Delta_{\mu_T}(i),$$

which happens when:

$$C_t(i) = \sqrt{\frac{\xi \log(N_t)}{N_t(i)}} \leq \sqrt{\frac{\xi \log(T)}{N_t(i)}} \leq \frac{1}{2}\Delta_{\mu_T}(i),$$

i.e., when:

$$N_t(i) \geq \left\lceil \frac{4\xi \log(T)}{(\Delta_{\mu_T}(i))^2} \right\rceil = A_i.$$

We will be pessimistic about the arm choices when $\{I_t = i \neq i_t^*, N_t(i) < A_i\}$ and we consider them being all wrong choices. We define $\mathcal{T}_i = \{t \in [W, T] | N_t(i) \geq A_i\}$. Remark that $N_t(i)$ contains also pulls in the interval $[0, W]$, i.e., the interval where samples come from distribution θ_0 and we are not interested in them when counting $\tilde{N}_{W:T}(i)$. We get:

$$\begin{aligned} Q_{W:T}(i) &\leq \sum_{t=W}^{T} \Bigg(\mathbb{1}_{\{\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\}} + \mathbb{1}_{\{\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*)\}} + \\ &\quad + \mathbb{1}_{\{\mu_t(i_t^*) - \mu_t(i) \leq 2C_t(i)\}} \Bigg) \\ &\leq (A_i - W) + \sum_{t \in \mathcal{T}_i} \left(\mathbb{1}_{\{\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\}} + \mathbb{1}_{\{\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*)\}} \right), \end{aligned}$$

where:

$$\bar{X}_t(i) = \sum_{s=1}^{t} \mathbb{1}_{\{i_s = i\}} X_s(i) = \sum_{s=1}^{W} \mathbb{1}_{\{i_s = i\}} X_s(i) + \sum_{s=W+1}^{t} \mathbb{1}_{\{i_s = i\}} X_s(i),$$

and $\bar{X}_t(i)$ and $\bar{X}_t(i_t^*)$ are biased estimators of the mean of distribution θ_1 since they contain samples from distribution θ_0. Taking the expectation over the experiments, we obtain:

$$\begin{aligned} \mathbb{E}[Q_{W:T}(i)] &\leq (A_i - W) + \\ &\quad + \sum_{t \in \mathcal{T}_i} \left(P\left(\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\right) + P\left(\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*)\right) \right) \end{aligned}$$

We upper bound the probability of $\{\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\}$ by separately considering:

- the bias $\frac{M_t(i)}{N_t(i)} - \mu_t(i)$
- the fluctuations of $\bar{X}_t(i)$ around $\frac{M_t(i)}{N_t(i)}$

where

$$N_t(i) = \sum_{s=1}^{t} \mathbb{1}_{\{i_s=i\}} = \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}} + \sum_{s=W+1}^{t} \mathbb{1}_{\{i_s=i\}},$$

$$M_t(i) = \sum_{s=1}^{t} \mathbb{1}_{\{i_s=i\}}\mu_s(i) = \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}}\mu_{\theta_0}(i) + \sum_{s=W+1}^{t} \mathbb{1}_{\{i_s=i\}}\mu_{\theta_1}(i).$$

We start with the bias. Since $\frac{M_t(i)}{N_t(i)}$ is a convex combination of $\mu_{\theta_0}(i)$ and $\mu_{\theta_1}(i)$, it belongs to the interval $[0, 1]$. Hence:

$$\left|\frac{M_t(i)}{N_t(i)} - \mu_t(i)\right| \leq 1.$$

Moreover, $\forall t \in \{W, ..., T\}$:

$$\begin{aligned}
|M_t(i) - \mu_t(i)N_t(i)| =& \left| \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}}\mu_{\theta_0}(i) + \sum_{s=W+1}^{t} \mathbb{1}_{\{i_s=i\}}\mu_{\theta_1}(i) - \right. \\
&\left. + \mu_{\theta_1}(i) \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}} - \mu_{\theta_1}(i) \sum_{s=W+1}^{t} \mathbb{1}_{\{i_s=i\}} \right| \\
=& \left| \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}}(\mu_{\theta_0}(i) - \mu_{\theta_1}(i)) + \sum_{s=W+1}^{t} \mathbb{1}_{\{i_s=i\}}(\mu_{\theta_1}(i) - \mu_{\theta_1}(i)) \right| \\
=& \left| \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}}(\mu_{\theta_0}(i) - \mu_{\theta_1}(i)) \right| \\
\leq& \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}}|\mu_{\theta_0}(i) - \mu_{\theta_1}(i)|.
\end{aligned}$$

We remark that, since the window W is the amount of history indicated by the CUSUM algorithm after a true alarm is raised, the samples inside must be, in some way, more probable to come from θ_1 than from θ_0, despite coming from θ_0. Thus, we call μ_W the mean of the samples inside W and we use it in the bound instead of μ_{θ_0}:

$$|M_t(i) - \mu_t(i)N_t(i)| \leq \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}}|\mu_W(i) - \mu_{\theta_1}(i)|$$

We must consider two cases:

- if $\mu_{\theta_1} \geq \mu_{\theta_0}$: then $2\epsilon \leq \mu_1 \leq 1$ and the CUSUM walk that detects increments in the mean will be triggered. We have:

$$\sum_{k\in[1,W]} (y_k) - W(\hat{\mu}_{\theta_0} + \epsilon) \geq h$$

and therefore

$$\mu_W = \mathbb{E}[y_k] \geq \frac{h}{W} + \hat{\mu_{\theta_0}} + \epsilon$$

thus

$$\epsilon + \frac{h}{W} \leq \mu_W \leq 1$$

  - if $\mu_W < \mu_{\theta_1}$, then:

$$\begin{aligned}
|M_t(i) - \mu_t(i)N_t(i)| &\leq \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}}(\mu_{\theta_1}(i) - \mu_W(i)) \\
&\leq W(\mu_{\theta_1}(i) - \mu_W(i)) \\
&\leq W\left(\mu_{\theta_1}(i) - \hat{\mu_{\theta_0}} - \epsilon - \frac{h}{W}\right) \\
&\leq W\left(1 - 0 - \epsilon - \frac{h}{W}\right) \\
&= W(1-\epsilon) - h
\end{aligned}$$

  - if $\mu_W \geq \mu_{\theta_1}$, then:

$$\begin{aligned}
|M_t(i) - \mu_t(i)N_t(i)| &\leq \sum_{s=1}^{W} \mathbb{1}_{\{i_s=i\}}(\mu_W(i) - \mu_{\theta_1}(i)) \\
&\leq W(\mu_W(i) - \mu_{\theta_1}(i)) \\
&\leq W(1-2\epsilon)
\end{aligned}$$

  Therefore, we get

$$|M_t(i) - \mu_t(i)N_t(i)| \leq \max(W(1-\epsilon) - h, W(1-2\epsilon)) \leq W(1-\epsilon)$$

- if $\mu_{\theta_1} < \mu_{\theta_0}$: then $0 \leq \mu_1 \leq 1 - 2\epsilon$ and the CUSUM walk that detects decreases in the mean will be triggered. We have:

$$-\sum_{k\in[1,W]} (y_k) + W(\hat{\mu}_{\theta_0} - \epsilon) \geq h$$

and therefore

$$\mu_W = \mathbb{E}[y_k] \leq \hat{\mu_{\theta_0}} - \epsilon - \frac{h}{W}$$

thus

$$0 \leq \mu_W \leq 1 - \epsilon - \frac{h}{W}$$

Considering both cases when $\mu_1 \geq \mu_W$ and when $\mu_1 < \mu_W$, we draw the same conclusions of the previous analysis:

$$|M_t(i) - \mu_t(i)N_t(i)| \leq W(1-\epsilon)$$

Summing up the previous results we have:

$$\left|\frac{M_t(i)}{N_t(i)} - \mu_t(i)\right| \leq \frac{W(1-\epsilon)}{N_t(i)},$$

which combined with

$$\left|\frac{M_t(i)}{N_t(i)} - \mu_t(i)\right| \leq 1$$

provides the following:

$$\left|\frac{M_t(i)}{N_t(i)} - \mu_t(i)\right| \leq \min\left(1, \frac{W(1-\epsilon)}{N_t(i)}\right).$$

Using the fact that:

$$min(1, x) \leq \sqrt{x}$$

we have:

$$\left|\frac{M_t(i)}{N_t(i)} - \mu_t(i)\right| \leq \sqrt{\frac{W(1-\epsilon)}{N_t(i)}}.$$

We want to make the bias smaller than half of the exploration bonus so that we can control it:

$$\sqrt{\frac{W(1-\epsilon)}{N_t(i)}} \leq \frac{1}{2}C_t(i) = \frac{1}{2}\sqrt{\frac{\xi \log(t)}{N_t(i)}}$$

and we obtain

$$W(1-\epsilon) \leq \frac{\xi}{4}\log(t),$$

which is true for

$$t \geq e^{\frac{4W(1-\epsilon)}{\xi}}.$$

We call:

$$\mathcal{G}(W) = e^{\frac{4W(1-\epsilon)}{\xi}}.$$

The other half of the exploration bonus is used to control the fluctuations:

$$\begin{aligned} P\left(\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\right) &\leq P\left(\bar{X}_t(i) \geq \mu_t(i) + \frac{1}{2}\sqrt{\frac{\xi \log(t)}{N_t(i)}} + \left|\frac{M_t(i)}{N_t(i)} - \mu_t(i)\right|\right) \\ &\leq P\left(\bar{X}_t(i) - \frac{M_t(i)}{N_t(i)} \geq \frac{1}{2}\sqrt{\frac{\xi \log(t)}{N_t(i)}}\right). \end{aligned}$$

Applying the Hoeffding's inequality, we obtain:

$$\begin{aligned} P\left(\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\right) &\leq P\left(\bar{X}_t(i) - \frac{M_t(i)}{N_t(i)} \geq \frac{1}{2}\sqrt{\frac{\xi \log(t)}{N_t(i)}}\right) \\ &\leq e^{-\frac{2N_t(i)\frac{1}{4}\xi\log(t)}{N_t(i)}} \\ &= e^{-\frac{\xi\log(t)}{2}} \\ &= t^{-\frac{\xi}{2}} \end{aligned}$$

Considering also the event $\{\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*)\}$, we obtain:

$$\begin{aligned} &\sum_{t=W}^{T}\left(P\left(\bar{X}_t(i) \geq \mu_t(i) + C_t(i)\right) + P\left(\bar{X}_t(i_t^*) \leq \mu_t(i_t^*) - C_t(i_t^*)\right)\right) \\ &\leq \sum_{t=W}^{T}\left(2t^{-\frac{\xi}{2}}\right) \leq 2\sum_{t=1}^{\infty} t^{-\frac{\xi}{2}} \leq 2\zeta\left(\frac{\xi}{2}\right) \end{aligned}$$

In conclusion, we have:

$$\mathbb{E}[Q_{W:T}(i)] \leq (A_i - W) + \mathbb{E}_W[\mathcal{G}(W)] + 2\zeta\left(\frac{\xi}{2}\right)$$

Substituting $Q_{W:T}(i)$ in the bound on $\tilde{N}_{W:T}(i)$, we obtain:

$$\mathbb{E}[\tilde{N}_{W:T}(i)] \leq (A_i - W) + \mathbb{E}_W[\mathcal{G}(W)] + 2\zeta\left(\frac{\xi}{2}\right) + (T - W)\frac{\alpha}{K}.$$

5.2.8 Analysis of the meta-algorithm with breakpoint prediction

We remark that, as explained in Section 5.2.1, using a generic stationary sub-policy $\mathfrak{U}_{SUB}$ and a generic CDT without breakpoint prediction, an upper bound on the pseudo-regret of the meta policy $\mathfrak{U}$ is:

$$\mathbb{E}[L_T(\mathfrak{U})] \leq (1 + \mathbb{E}[F] + \gamma_T)\mathbb{E}[L_T(\mathfrak{U}_{SUB})] + \gamma_T\left(\mathbb{E}[D]\frac{K}{\alpha}\right)$$

In Section 5.2.7, the behaviour of a stationary bandit policy $\mathfrak{U}$, namely UCB1-Explore, on a ST-MAB with the first W samples coming from a CUSUM breakpoint prediction has been analysed. We call the obtained upper bound on the expected regret $\mathbb{E}[L_T^W(\mathfrak{U})]$. Summing up, the CUSUM procedure suggests $W \geq \frac{h}{1-\epsilon}$ past samples, making the policy choose a sub-optimal arm on average at most $\mathbb{E}[\mathcal{G}(W)]$ timesteps. Despite simulations on the CUSUM CDT in Section 2.4.3 show that both the probability of over-shooting and the prediction error are low in practice, we were not able to find

an upper bound on the probability of overshoot, since it depends on multiple factors like the detection point, the distributions and all the CUSUM parameters. When breakpoint prediction is used, an upper bound on the expected pseudo-regret of the meta policy can be easily found by considering the worst case in which the worst overshoot always happens. Thus, the upper bound is the following:

$$\mathbb{E}[L_T(\mathfrak{U})] \leq (1 + \mathbb{E}[F] + \gamma_T)\mathbb{E}[L_T^W(\mathfrak{U}_{SUB})] + \gamma_T \left(\mathbb{E}[D]\frac{K}{\alpha}\right).$$

where $L_T^W(\mathfrak{U}_{SUB})$ is an upper bound on the regret achieved by the subpolicy $\mathfrak{U}_{SUB}$ on a stationary MAB with the first W samples coming from a CUSUM breakpoint prediction.

5.2.9 Imposing a maximum amount of history given by the breakpoint prediction procedure

In order to constrain the bias that may be introduced by a breakpoint prediction, as previously mentioned in Section 5.2.8, since the distribution of the breakpoint predictions of the CUSUM algorithm is unknown, we limit the amount of history to be included in the next phase execution of the algorithm. We define the maximum amount of history:

$$W^* = \frac{\xi}{8}\log(T\log(T)),$$

so that the bias becomes smaller than half of the exploration bonus on average after the following amount of timesteps:

$$\mathbb{E}_W[\mathcal{G}(W)] \leq \mathbb{E}_{W=W^*}[\mathcal{G}(W)] = (T\log(T))^{\frac{(1-\epsilon)}{2}}.$$

5.2.10 Expected regret of CUH

Let $\tilde{N}_{T,\mathfrak{U}_{CUH}}(i)$ be the number of times policy CUH selects the sub-optimal arm i up to time T on a ANS-MAB setting. Let $\tilde{N}^W_{T,\mathfrak{U}_{SUB}}(i)$ be the number of times policy UCB1-Explore selects the sub-optimal arm i up to time T on a ST-MAB setting with W initial observations coming from a CUSUM breakpoint prediction. As studied in the analysis of the meta-algorithm with breakpoint prediction in Section 5.2.8, combining with the analysis of the UCB1-Explore policy in a stationary setting with bias in Section 5.2.7, choosing CUSUM as CDT and using the bounds on the expected delay and expected number of false alarms shown in Section 5.2.4, fixing the maximum amount of past samples to consider as in Section 5.2.9, the expected pseudo-regret of the CUH policy $\mathfrak{U}$ on a ANS-MAB setting has the following upper

bound:

$$\mathbb{E}[\tilde{N}_{T,\mathfrak{U}_{CUH}}(i)] \leq (1+\mathbb{E}[F]+\gamma_T)\mathbb{E}[\tilde{N}^W_{T,\mathfrak{U}_{SUB}}(i)] + \gamma_T\left(\mathbb{E}[D]\frac{K}{\alpha}\right)$$
$$\leq (1+\mathbb{E}[F]+\gamma_T)\left(A_i + \mathbb{E}_{W=W^*}[\mathcal{G}(W)] + 2\zeta\left(\frac{\xi}{2}\right) + T\frac{\alpha}{K}\right) + \gamma_T\left(\mathbb{E}[D]\frac{K}{\alpha}\right)$$
$$\leq (1+\mathbb{E}[F]+\gamma_T)\left(A_i + \sqrt{T\log(T)}^{(1-\epsilon)} + 2\zeta\left(\frac{\xi}{2}\right) + T\frac{\alpha}{K}\right) + \gamma_T\left(\mathbb{E}[D]\frac{K}{\alpha}\right),$$

where:

$$A_i = \left\lceil \frac{4\xi\log(T)}{(\Delta_{\mu_T}(i))^2} \right\rceil,$$
$$\mathbb{E}[D] \leq \left(\log(3) + \frac{2}{\lambda}e^{-2\epsilon^2,M}\right)(h+1)$$
$$\mathbb{E}[F] \leq \frac{2T}{\left(1-2e^{-2\epsilon^2 M}\right)e^{C_1 h}}.$$

Choosing $h = \frac{1}{C_1}\log\left(\frac{T}{\gamma_T}\right)$ and $\alpha = K\sqrt{\frac{C_2\gamma_T}{C_1 T}\log\left(\frac{T}{\gamma_T}\right)}$, the upper bound becomes:

$$\mathbb{E}[\tilde{N}_{T,\mathfrak{U}_{CUH}}(i)] = O\left(\gamma_T\left(A_i + \sqrt{T\log(t)}^{1-\epsilon} + \sqrt{\frac{C_2}{C_1}T\gamma_T\log\left(\frac{T}{\gamma_T}\right)}\right) + \gamma_T\sqrt{\frac{C_1 T}{C_2\gamma_T}\log\left(\frac{T}{\gamma_T}\right)}\right)$$
$$= O\left(\sqrt{T\gamma_T\log\left(\frac{T}{\gamma_T}\right)}\right).$$

5.3 Discussion

The contribution provided in this chapter is two-fold. First, we prove a regret upper bound for an actively adaptive meta-policy which is independent from the CDT and the stationary subpolicy used, provided that an upper bound of the number of rounds required to gather $\mathbb{E}[D]$ samples on a suboptimal arm is specified. Second we derive a regret upper bound for the CUH policy when the maximum amount of history is fixed, which is the first result, up to our knowledge, concerning a MAB policy using breakpoint prediction. The obtained results concern the use of breakpoint prediction procedures when coupling stationary policies with a CDT by providing, considering the worst possible bias, an upper bound on the expected regret which is not larger than the one obtained without using breakpoint prediction procedures.

Chapter 6

Experiments

In this chapter, we test our novel policy CUH with state-of-the-art algorithms for the NS-MAB setting on both synthetic and real-world datasets and we discuss the results. The synthetic scenarios have been devised to cover several situations present in common ANS-MAB problems, showing both the advantages and the limitations of our proposed algorithm.

6.1 Experimental setting

In the following experiments, we test:

- stationary policies: UCB1;
- passively adaptive policies: D-UCB and SW-UCB;
- actively adaptive policies: Adapt-EvE, CUSUM-UCB and two instances of CUH: one with fixed maximum history, that we call CUSUM-UCB with Logarithmic History (CU-LH), one with unbounded maximum history, that we call CUSUM-UCB with Unbounded History (CU-UH).

Each policy has been designed for a specific setting in mind, which has been reported in Figure 6.1. The parameters chosen for each policy can be found in Table 6.1. Note that, for both the CUSUM-UCB and the CUH policies, the parameters α and h used are different from the ones in Corollary 5.1, since they minimize an upper bound on the regret, rather than the regret itself. Moreover, we consider a larger number of past samples given by the upper bound because, in the experiment, the time horizon is not sufficiently long to make W^* large enough to differentiate the performance of CUH and CUSUM-UCB. As a consequence, we try two different amounts of maximum history: a logarithmic choice of $W^* = \log\left(\frac{T}{\Upsilon_T}\right)$ past samples, and an unbounded choice $W^* = \infty$, with the goal to see, in practice, if the bias introduced by all the past samples suggested by the CDT is dramatically influencing the

policy. Note that M and ϵ are obtained from prior knowledge of the datasets. The policies will be tested in five different scenarios. Each experiments is run 100 times and we store the average pseudo-regret, along with its 95% two-sided confidence interval.

In all the scenarios, the minimum amount of change of mean of the distribution of an arm is set to 0.05, thus $\epsilon = 0.025$ and M is set to 30, in order to have detection delays which are not too long with respect to the time horizon. Indeed, as shown in Section 5.2.4, the expected delay of the CUSUM procedure is upper bounded by:

$$\mathbb{E}[D] \leq \left(\log(3) + \frac{2}{\lambda} e^{-2\epsilon^2 M}\right)(h+1),$$

where λ is the minimal non-trivial gap between expectation and closest grid point of any arm, as explained in Section 3.1. Since $M = 30$, each grid slot measures $\frac{1}{30} = 0.0\bar{3}$. Since, in our experiments, we generate probabilities up to 3 decimal slots, the smallest non-trivial difference between a generated probability and a grid border is $0.0\bar{3} - 0.033 = 0.000\bar{3} = \lambda$. Finally, since we set the threshold $h = \log\left(\frac{T}{\Upsilon_T}\right)$, the expected detection delay is upper bounded by nearly $4.5\,10^4$, which is smaller than the shortest time horizon used in the experiments, i.e., $T = 10^5$.

The minimum difference between the mean of the optimal arm and the mean of a suboptimal arm Δ_{μ_T} is set to 0.03, a value large enough to make the the first A_i samples of the UCB1-Explore subpolicy, defined in Section 5.2.2, less than the shortest time horizon of our experiments:

$$A_i := \frac{4\xi \ln(T)}{(\Delta_{\mu_T})^2} \approx 10^4 \leq \min(T) = 10^5.$$

Scenario 1. The first scenario contains experiments whose aim is to see how the algorithms react when they face a single change on the optimal arm or a change on a single suboptimal arm. Some policies, like CUSUM-UCB and the variants of CUH, have ad-hoc uniform exploration to pull sub-optimal arms enough times to detect possible breakpoints, while Adapt-EvE does not. All the experimental settings have two arms, where one is stationary with mean 0.5 and the other has a breakpoint at half of the time horizon $T/2$, having mean reward $0.5 \pm \delta$ before the breakpoint and mean reward $0.5 \mp \delta$ after, where δ is the change amount parameter of the setting. The configurations are determined by the set of the following parameters:

- time horizon T;
- change type: the optimal arm changes distribution and becomes sub-optimal, or the sub-optimal arm changes distribution and becomes optimal;

Policy	Parameters	Selection
UCB1	No parameters	
D-UCB	$\gamma = 1 - 1/(4B)\sqrt{\frac{\Upsilon_T}{T}}$	lowest upper bound on regret
SW-UCB	$\tau = 2B\sqrt{\frac{T\log(T)}{\Upsilon_T}}$	lowest upper bound on regret
Adapt-EvE	$M = 1000$ CDT: PH test with $\delta = 0.005$ and $h = 80$	suggested by the authors
CUSUM-UCB	$h = \log(\frac{T}{\Upsilon_T})$ $\alpha = \sqrt{\frac{\Upsilon_T}{T}\log(\frac{T}{\Upsilon_T})}$	suggested by the authors
CU-LH	$h = \log(\frac{T}{\Upsilon_T})$ $\alpha = \sqrt{\frac{\Upsilon_T}{T}\log(\frac{T}{\Upsilon_T})}$ $W^* = \log\left(\frac{T}{\Upsilon_T}\right)$	suggested by the authors
CU-UH	$h = \log(\frac{T}{\Upsilon_T})$ $\alpha = \sqrt{\frac{\Upsilon_T}{T}\log(\frac{T}{\Upsilon_T})}$ $W^* = \infty$	suggested by the authors

Table 6.1: Values of the parameters of the policies used in the experiments. For Adapt-EvE, since no theoretical analysis is available, parameters suggested by the authors are used. For CUSUM-UCB and the variants of CUH, despite a theoretical analysis that minimizes an upper bound on the pseudo-regret is available, the authors suggest this choice of parameters that better focuses on the minimization of the regret.

parameter	possible values
T	10^5, 10^6
change type	optimal arm changes, sub-optimal arm changes
change amount δ	0.1, 0.2, 0.4

Table 6.2: Possible values of each parameter in Scenario 1.

- change amount: how much the mean of the non-stationary arm differs from the mean of the stationary arm.

The possible values of each parameter are listed in Table 6.2.

Scenario 2. The second scenario tests the algorithms with equally-spaced breakpoints. Every time there is a breakpoint, the best arm changes, therefore, it is always useful if one wants to control the regret to detect a change. The configurations are decided according to the following parameters:

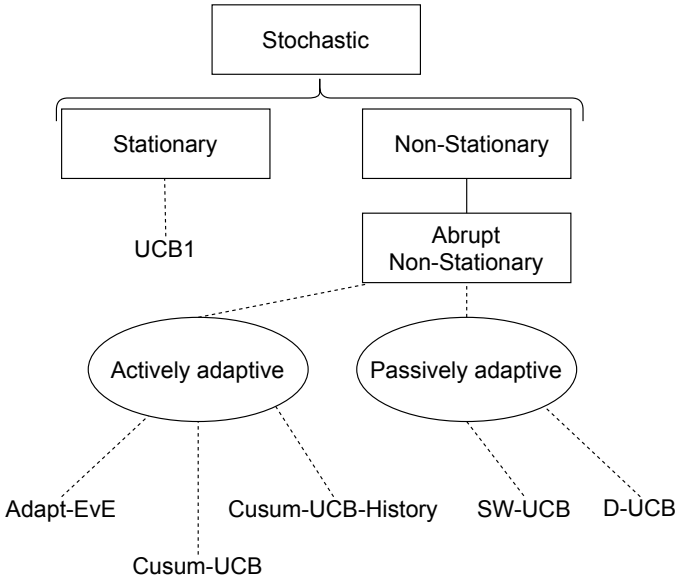


Figure 6.1: Algorithms used in the experiments, along with the settings they have been designed for and their categorization.

- time horizon T;
- number of breakpoints: a high number of breakpoints means that stationary phases are smaller, thus requiring the policies to be fast in detecting changes;
- number of arms: as the number of arms increases, the policies must explore sub-optimal arms more to detect changes on them;
- switching: global switching or per-arm switching. In a global switching setting, the distributions of the rewards of all the arms change at each breakpoint. In a per-arm switching setting, only one distribution changes at each breakpoint.

The possible values of each parameter are listed in Table 6.3. Since the results depend strongly from the specific experiment realization of the parameters, we make each configuration generate 5 different experiments, we test each experiment 100 times and we average the results over the generated experiments.

Scenario 3. The third scenario contains experiments where the optimal arm distribution does not change and so the optimal arm remains always the same. The goal is to see the downsides of focusing attention on sub-optimal arm changes and to understand if breakpoint prediction helps. The configurations are decided according to the following parameters:

parameter	possible values
T	10^5, 10^6
number of breakpoints	1, 8
number of arms	2, 8
switching	global switching, per-arm switching

Table 6.3: Possible values of each parameter in Scenario 2.

parameter	possible values
T	10^5
number of breakpoints	2, 4, 8, 24
number of arms	4, 8

Table 6.4: Possible values of each parameter in Scenario 3.

- number of breakpoints;
- number of arms.

We present experiments with time horizon of $T = 10^5$ timestep only, since we noticed that very similar results are obtained with different time horizons. The possible values of each parameter are listed in Table 6.4.

Scenario 4. The fourth scenario contains settings in which the magnitude of each change in the mean of a distribution is bounded: the goal is to see if breakpoint prediction, in presence of bounded changes, introduces less bias in the estimates. We further add the constraint that there exists at least one breakpoint in which a suboptimal arm becomes the optimal arm, so that a policy must change the arm selection at least once. The configurations are decided according to the following parameters:

- number of breakpoints;
- number of arms;
- maximum mean change on an arm after a breakpoint.

The possible values of each parameter are listed in Table 6.5. As in the Scenario 2, since the results depend on the specific experiment realization of the parameters, we make each configuration generate 5 different experiments, we test each experiment 100 times and we average the results over the generated experiments.

parameter	possible values
T	10^5
number of breakpoints	1, 8, 24
number of arms	2, 8
maximum mean change	0.1, 0.15, 0.2

Table 6.5: Possible values of each parameter in Scenario 4.

Scenario 5. The fifth scenario involves real-world data from a Yahoo! dataset [42]. This dataset contains information on the click through rate of articles displayed in the Featured Tab of the Today Module on Yahoo!'s front page. It contains 15 days of data from October 2 to 16, 2011. We choose three random days from them and we run a separate experiment for each of these days. For each day independently, we select the 5 most occurring articles and we consider as time interval the one having as birth the latest first occurrence of an article, and having as death the earliest last occurrence of an article. With these operations, we often find a time interval long about 10^5 timesteps for each day on the dataset. Furthermore, we divide this time interval into M equally-long time slots and we compute the mean click through rate of each arm in each time slot, obtaining, for each article, a non-stationary Bernoulli distribution with $M-1$ breakpoints. We use these distributions in our experiment and we observe the results.

6.2 Results

This section contains the results of the experiments described in Section 6.1. The outcomes are organized in tables and we mark in bold the results which are significantly better than the others. Multiple results are in bold when their confidence intervals overlap.

6.2.1 Scenario 1: optimal arm changes versus sub-optimal arm changes

This scenario is about investigating the behaviour of the selected algorithms when they face changes in the optimal arm, which is pulled frequently, or changes in the sub-optimal arm, which is pulled few times. CUSUM-UCB and the variants of CUH have been designed with an uniform exploration over the not optimal arms in order to detect breakpoints on them, while Adapt-EvE has not such induced exploration. The values of the parameters of the environments can be found in Table 6.2. The following settings are of interest.

Setting 1. Consider the setting in which the optimal arm changes and be-

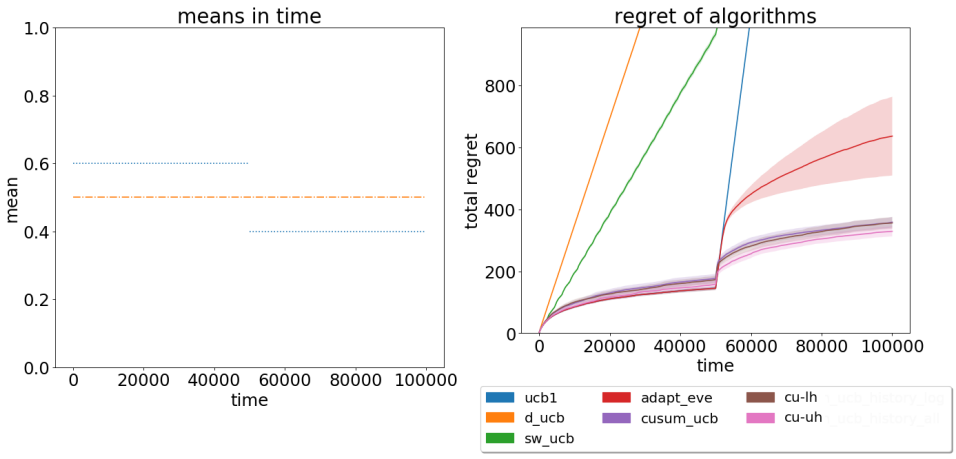


Figure 6.2: Scenario 1. Environment in which the optimal arm changes and becomes a sub-optimal arm with $T = 10^5$ and $\delta = 0.1$. 95% confidence bound on the pseudo-regret of each policy are shown.

comes a sub-optimal arm, with $T = 10^5$ and $\delta = 0.1$, as shown in Figure 6.2. In this case, UCB1 fails to adapt to the change of the optimal arm after the changepoint, because its estimate of the mean reward has a lot of inertia (i.e., the first arm has been pulled a linear amount of times in T before the changepoint) and, thus, it is updated slowly after the change. Non-stationary passively adaptive policies perform better than standard UCB1, but fail to reach the performance of non-stationary actively adaptive policies, which are able to effectively split the time horizon into separate stationary bandit problems by successfully locating the changepoint and resetting themselves. Among these, CUSUM-UCB and the variants of CUH achieve the lowest regret, as they reset only the information on the arm whose distribution has changed, thus doing better than Adapt-EvE. In particular, CU-UH performs slightly better than CU-LH, suggesting that past samples may be helpful in this setting.

Setting 2. Consider the setting in which the sub-optimal arm changes and becomes the new optimal arm, with $T = 10^5$ and $\delta = 0.1$, as shown in Figure 6.3. UCB1 is good at detecting changes in sub-optimal arms in this case because, when it samples the second arm after the changepoint, its estimate of the mean of the sub-optimal arm has small inertia (i.e., the sub-optimal arm has been pulled a few times before the changepoint), thus it is able to update the expected reward estimate rather fast. Adapt-EvE, since its CDT monitors only decreases in the mean, does not reset and thus performs similarly to UCB1. CUSUM-UCB and the variants of CUH achieve higher regret because detecting changes on sub-optimal arms takes some time and the mean of the sub-obtimal arm has more inertia than the one of UCB1,

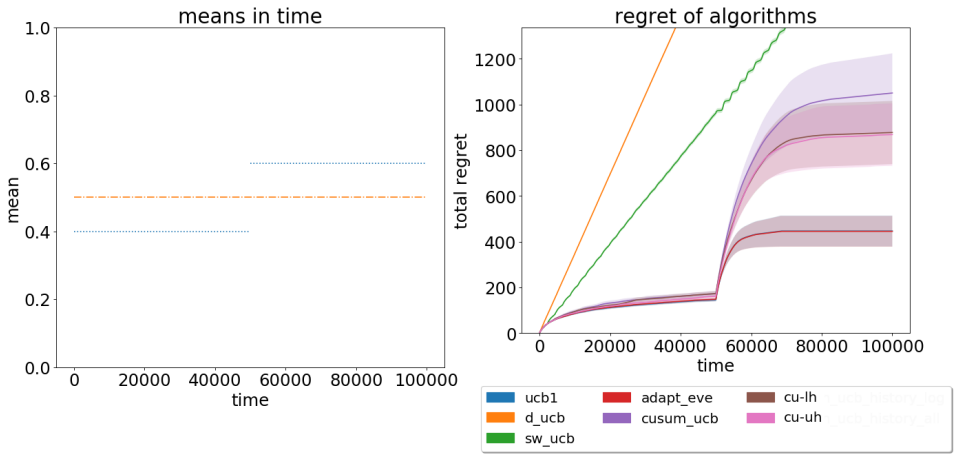


Figure 6.3: Scenario 1. Environment in which the sub-optimal arm changes and becomes the new optimal arm with $T = 10^5$ and $\delta = 0.1$. 95% confidence bound on the pseudo-regret of each policy are shown.

because of the random exploration. Again, as shown in Table 6.7, past samples seem to be helpful, but does not provide a statistically significant improvement in terms of regret.

Considering the same settings, but with higher T and δ, gives different results.

Setting 3. Consider the setting in which the optimal arm changes and becomes a sub-optimal arm, with $T = 10^6$ and $\delta = 0.4$, as shown in Figure 6.4. All the policies behave similarly to the case depicted in Figure 6.2, thus T and δ do not affect significantly the performance of these algorithms in this setting.

Setting 4. Consider the setting in which the sub-optimal arm changes and becomes the optimal arm, with $T = 10^6$ and $\delta = 0.4$, as shown in Figure 6.5. As T and δ increase, UCB1 is less and less able to adjust the mean of the sub-optimal arm in a few rounds, even if it has small inertia, and achieves regret larger than CUSUM-UCB and the variants of CUH ones. Past history seems to provide no advantage in this case.

An overall overview of the results presented bebore is provided in Tables 6.6 and 6.7. The outcomes do not suggest any improvement or decline of CU-LH with respect to CUSUM-UCB, but in two experiments there is statistical evidence of CU-UH being better than CUSUM-UCB, both in a setting in which the optimal arm becomes sub-optimal. In every setting in which the optimal arm becomes suboptimal, CU-UH achieves lower average regret than CU-LH and CUSUM-UCB, but the difference is not always significant. When the sub-optimal arm becomes the optimal arm, the confidence

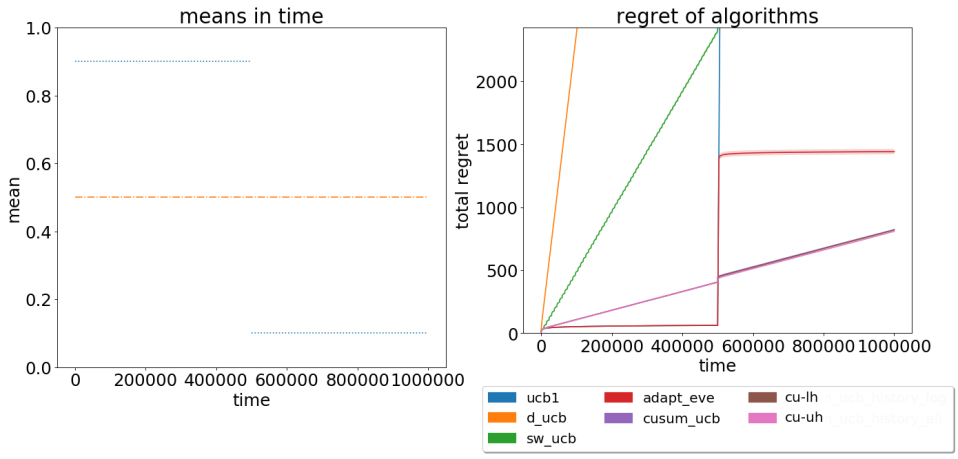


Figure 6.4: Scenario 1. Environment in which the optimal arm changes and becomes a sub-optimal arm with $T = 10^6$ and $\delta = 0.4$. 95% confidence bound on the pseudo-regret of each policy are shown.

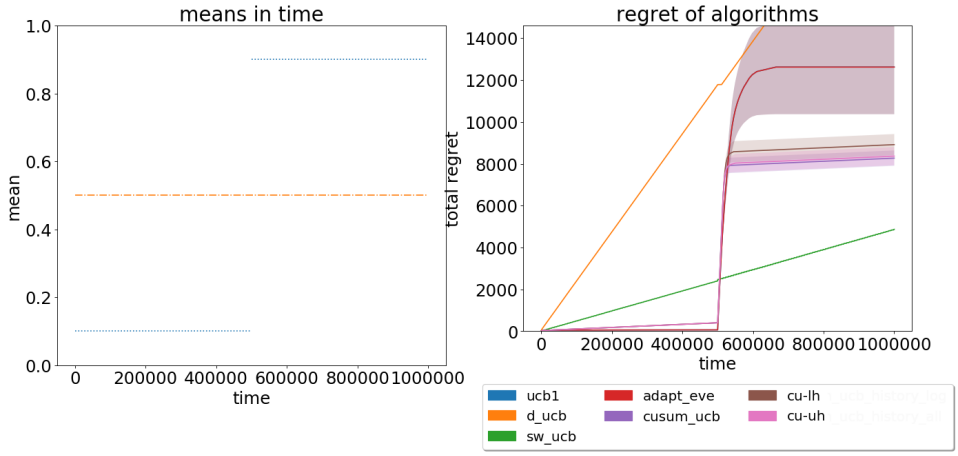


Figure 6.5: Scenario 1. Environment in which the sub-optimal arm changes and becomes the optimal arm with $T = 10^6$ and $\delta = 0.4$. 95% confidence bound on the pseudo-regret of each policy are shown.

intervals of the three policies CUSUM-UCB, CU-LH and CU-UH always intersect. In general, we see that, when a breakpoint produces a change in the optimal arm, breakpoint prediction does not produce a significant increase or decrease of regret, regardless of which arm changes.

	$T = 10^5$			$T = 10^6$		
	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.2$
UCB1	1622 ± 20	3483 ± 30	7683 ± 47	18364 ± 104	40164 ± 167	86517 ± 209
D-UCB	2101 ± 5	3465 ± 10	4704 ± 13	18468 ± 25	26605 ± 39	29202 ± 45
SW-UCB	1579 ± 11	1952 ± 14	1812 ± 13	10944 ± 43	10580 ± 44	8128 ± 34
Adapt-EvE	826 ± 82	636 ± 127	379 ± 10	5970 ± 1334	2993 ± 1307	1601 ± 36
CUSUM-UCB	$\mathbf{485 \pm 44}$	$\mathbf{359 \pm 17}$	237 ± 5	$\mathbf{940 \pm 86}$	$\mathbf{624 \pm 28}$	540 ± 6
CU-LH	$\mathbf{468 \pm 41}$	$\mathbf{356 \pm 19}$	235 ± 5	$\mathbf{937 \pm 91}$	649 ± 34	541 ± 8
CU-UH	$\mathbf{434 \pm 37}$	$\mathbf{329 \pm 16}$	$\mathbf{221 \pm 5}$	$\mathbf{881 \pm 84}$	$\mathbf{579 \pm 20}$	$\mathbf{526 \pm 6}$

Table 6.6: Scenario 1. 95% confidence bound on the pseudo-regret obtained considering change type where the optimal arm distribution changes and becomes a sub-optimal arm.

	$T = 10^5$			$T = 10^6$		
	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.2$
UCB1	$\mathbf{339 \pm 35}$	$\mathbf{447 \pm 68}$	$\mathbf{670 \pm 116}$	$\mathbf{1552 \pm 200}$	$\mathbf{3866 \pm 876}$	$\mathbf{5962 \pm 951}$
D-UCB	2090 ± 5	3431 ± 9	4574 ± 12	18451 ± 26	26430 ± 35	28693 ± 41
SW-UCB	1566 ± 11	1915 ± 15	1735 ± 12	10914 ± 38	10413 ± 44	7715 ± 29
Adapt-EvE	$\mathbf{342 \pm 35}$	$\mathbf{446 \pm 67}$	$\mathbf{666 \pm 116}$	$\mathbf{1500 \pm 195}$	$\mathbf{3668 \pm 860}$	$\mathbf{6719 \pm 1172}$
CUSUM-UCB	1381 ± 168	1051 ± 173	1567 ± 131	4980 ± 679	6450 ± 712	8590 ± 1108
CU-LH	1343 ± 164	878 ± 138	1582 ± 140	5426 ± 690	6099 ± 649	7988 ± 926
CU-UH	1313 ± 171	869 ± 136	1595 ± 142	6102 ± 832	6173 ± 694	9203 ± 1328

Table 6.7: Scenario 1. 95% confidence bound on the pseudo-regret obtained considering change type where the sub-optimal arm distribution changes and becomes the optimal arm.

6.2.2 Scenario 2: optimal arm always changes

This scenario is about investigating the behaviour of the selected algorithms with respect to a scenario in which, at each breakpoint, a sub-optimal arm becomes the new optimal arm. The parameters varying in this experiment are the time horizon, the number of breakpoints, the number of arms and the distinction between global switching and per-arm switching environments. The possible values of each parameter can be found in Table 6.3.

Typically, in global switching scenarios, the algorithms performance is similar to the one depicted in Figure 6.7, where Adapt-EvE performs better than CUSUM-UCB and the variants of CUH thanks to its global switching nature. The opposite happens in per-arm switching scenarios, where the algorithms performance is similar to the one in Figure 6.6.

Despite providing good results in most cases, there are cases in which

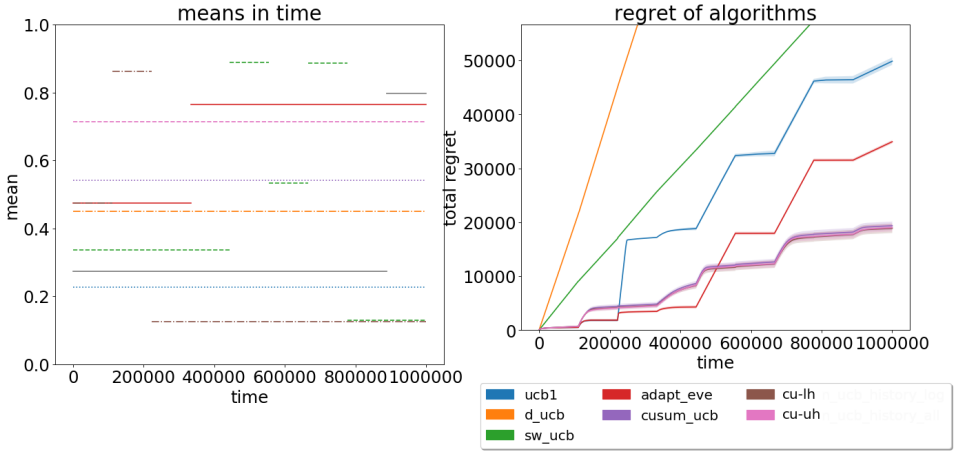


Figure 6.6: Scenario 2. Per-arm switching setting.

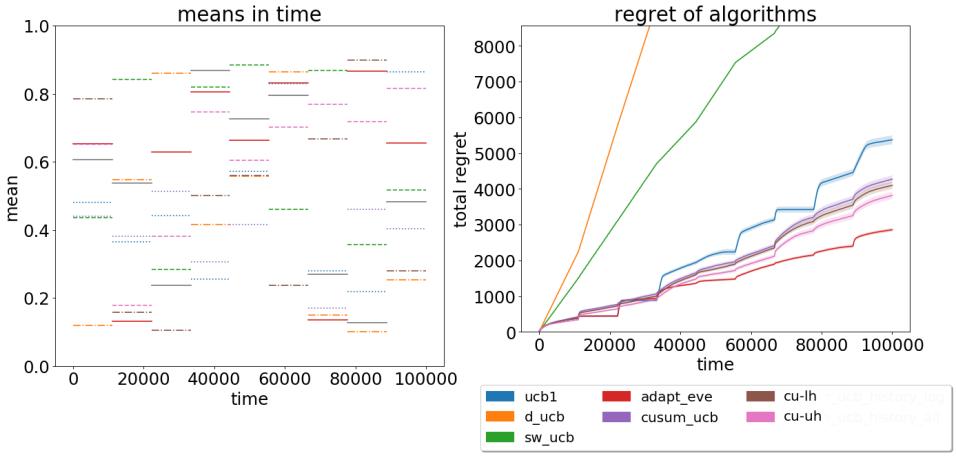


Figure 6.7: Scenario 2. Global switching setting.

Adapt-EvE performs really bad. In the setting in Figure 6.8, at the same timestep, both the optimal arm and the sub-optimal arm have an increase in the mean, resulting in the previous sub-optimal arm being optimal. The stationary sub-policy keeps choosing the arm depicted in blue right after the change because its upper bound moves upwards faster than the one of the arm depicted in orange, which is not sampled for a long time. In the setting in Figure 6.9, for two straight breakpoints, the sub-optimal arm becomes the optimal arm. The stationary sub-policy is able to react to the first change only, since the mean of the sub-optimal arm has small inertia. Once the second change happens, the mean of the sub-optimal arm is composed by a large number of samples since it has been an optimal arm in the past, and, thus, is able to adapt only slowly. The CUSUM-UCB policy and the variants of CUH perform well in all the settings, not showing any pathological case.

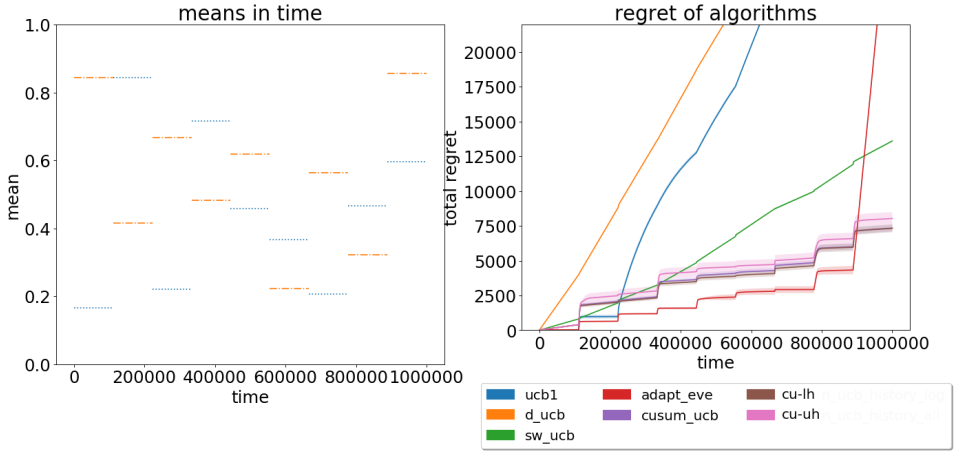


Figure 6.8: Scenario 2. Adapt-EvE first pathological case.

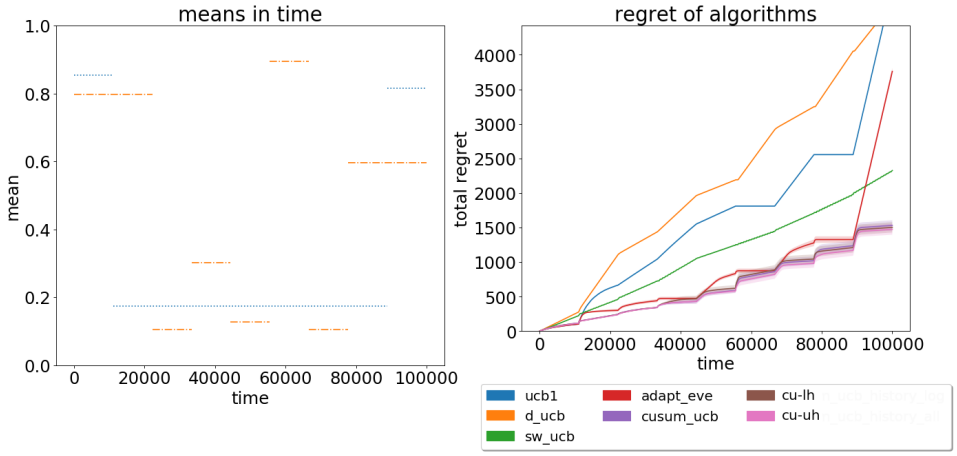


Figure 6.9: Scenario 2. Adapt-EvE second pathological case.

Overall results can be found in Tables 6.8, 6.9, 6.10 and 6.11. From the outcomes, we see that the confidence intervals of the policies CUSUM-UCB, CU-LH and CU-UH almost always intersect, thus concluding that, when the optimal arm changes at each breakpoint, breakpoint prediction does not produce any significant increase or decrease of regret.

	$T = 10^5$		$T = 10^6$	
	global	per-arm	global	per-arm
UCB1	2823 ± 33	3890 ± 80	27953 ± 112	21181 ± 308
D-UCB	4363 ± 12	4265 ± 10	25012 ± 35	23490 ± 33
SW-UCB	1532 ± 10	1702 ± 11	7528 ± 28	8476 ± 32
Adapt-EvE	$\mathbf{290 \pm 18}$	$\mathbf{600 \pm 71}$	2429 ± 481	4330 ± 1035
CUSUM-UCB	780 ± 45	$\mathbf{632 \pm 39}$	$\mathbf{1564 \pm 85}$	$\mathbf{1833 \pm 206}$
CU-LH	786 ± 45	$\mathbf{615 \pm 37}$	$\mathbf{1576 \pm 83}$	$\mathbf{1730 \pm 116}$
CU-UH	773 ± 43	$\mathbf{616 \pm 37}$	$\mathbf{1524 \pm 88}$	$\mathbf{1876 \pm 239}$

Table 6.8: Scenario 2. 95% confidence bound on the pseudo-regret when optimal arm always changes, 2 arms and 1 breakpoint.

	$T = 10^5$		$T = 10^6$	
	global	per-arm	global	per-arm
UCB1	2488 ± 40	4342 ± 39	23362 ± 541	11633 ± 566
D-UCB	18987 ± 15	18351 ± 13	116662 ± 57	99568 ± 47
SW-UCB	8172 ± 21	8436 ± 21	39547 ± 58	31570 ± 47
Adapt-EvE	$\mathbf{865 \pm 68}$	2599 ± 214	23061 ± 486	$\mathbf{4825 \pm 993}$
CUSUM-UCB	1851 ± 99	$\mathbf{1261 \pm 32}$	$\mathbf{7570 \pm 454}$	$\mathbf{6007 \pm 477}$
CU-LH	1865 ± 109	$\mathbf{1258 \pm 33}$	$\mathbf{7426 \pm 473}$	$\mathbf{6067 \pm 506}$
CU-UH	1683 ± 100	$\mathbf{1227 \pm 32}$	9186 ± 1052	$\mathbf{5943 \pm 485}$

Table 6.9: Scenario 2. 95% confidence bound on the pseudo-regret when optimal arm always changes, 8 arms and 1 breakpoint.

	$T = 10^5$		$T = 10^6$	
	global	per-arm	global	per-arm
UCB1	6159 ± 85	3988 ± 33	84197 ± 117	59281 ± 191
D-UCB	5807 ± 9	4795 ± 7	37720 ± 32	38654 ± 32
SW-UCB	2569 ± 11	2517 ± 10	12924 ± 31	14479 ± 34
Adapt-EvE	3006 ± 93	4441 ± 103	21159 ± 2157	33454 ± 909
CUSUM-UCB	$\mathbf{2065 \pm 76}$	$\mathbf{1632 \pm 70}$	$\mathbf{5728 \pm 216}$	$\mathbf{6637 \pm 284}$
CU-LH	$\mathbf{2029 \pm 78}$	$\mathbf{1603 \pm 73}$	$\mathbf{5672 \pm 213}$	$\mathbf{6670 \pm 288}$
CU-UH	$\mathbf{2164 \pm 97}$	$\mathbf{1600 \pm 74}$	6779 ± 587	$\mathbf{6661 \pm 324}$

Table 6.10: Scenario 2. 95% confidence bound on the pseudo-regret when optimal arm always changes, 2 arms and 8 breakpoints.

	$T = 10^5$		$T = 10^6$	
	global	per-arm	global	per-arm
UCB1	6027 ± 115	5979 ± 57	59373 ± 426	66384 ± 561
D-UCB	24879 ± 11	24588 ± 11	183728 ± 53	176934 ± 47
SW-UCB	12923 ± 19	13298 ± 21	69796 ± 61	70637 ± 61
Adapt-EvE	$\mathbf{4495 \pm 113}$	$\mathbf{4135 \pm 74}$	$\mathbf{14312 \pm 908}$	25962 ± 778
CUSUM-UCB	$\mathbf{4978 \pm 129}$	$\mathbf{4016 \pm 122}$	$\mathbf{13709 \pm 427}$	$\mathbf{17780 \pm 727}$
CU-LH	$\mathbf{4837 \pm 123}$	$\mathbf{3994 \pm 121}$	$\mathbf{13635 \pm 460}$	$\mathbf{17527 \pm 760}$
CU-UH	$\mathbf{4740 \pm 131}$	$\mathbf{3974 \pm 124}$	$\mathbf{14375 \pm 864}$	$\mathbf{18200 \pm 849}$

Table 6.11: Scenario 2. 95% confidence bound on the pseudo-regret when optimal arm always changes, 8 arms and 8 breakpoints.

arms	2		8	
breakpoints	2	4	2	4
UCB1	**173 ± 4.4**	**370 ± 9.6**	**425 ± 8.0**	**506 ± 8.1**
D-UCB	13736 ± 16.3	11868 ± 10.9	23972 ± 15.1	23208 ± 12.5
SW-UCB	4489 ± 17.2	5840 ± 15.7	9638 ± 20.1	12163 ± 22.8
Adapt-EvE	**174 ± 4.5**	**371 ± 9.8**	**425 ± 8.0**	**506 ± 8.1**
CUSUM-UCB	679 ± 16.1	984 ± 26.4	1058 ± 15.9	1607 ± 61.6
CU-LH	675 ± 13.9	943 ± 20.2	1047 ± 13.9	1624 ± 62.5
CU-UH	626 ± 9.3	881 ± 22.0	1013 ± 11	1286 ± 41.0

Table 6.12: Scenario 3. 95% confidence bound on the pseudo-regret when there are 2 or 4 breakpoints.

6.2.3 Scenario 3: optimal arm does not change

This scenario is about investigating the behaviour of the selected algorithms when the optimal arm is always the same and does not experience any non-stationarity throughout the experiment, but the other arms do. The goal is to see how much the change point detection procedure applied to non-optimal arms hinder the policies in achieving low regret and observe if breakpoint prediction helps in such a case.

In Figure 6.10, we show an example of a setting of this scenario: we discuss here only one setting because the policies behave very similarly in all the configurations. UCB1 performs well since the optimal arm is always the same. The same applies to Adapt-EvE, which does not have a uniform exploration of sub-optimal arms and, in practice, detects only decreases in the mean of the optimal arm, which is stationary in this scenario. Instead, CUSUM-UCB and the variants of CUH achieve higher regret because they explore sub-optimal arm for changes that would make the optimal arm different, without finding any. In particular, in this scenario CU-UH always performs significantly better than CU-LH and CUSUM-UCB: indeed, bias is not particularly harmful in this scenario, since all the past samples always support the hypothesis that the arm is sub-optimal and no precise estimates are needed.

Overall results are shown in Tables 6.12 and 6.13. Note how the advantage given by breakpoint prediction with respect to CUSUM-UCB increases as the number of arms and breakpoints increases. In all the experiments, CU-UH achieves a significantly lower pseudo-regret than the one of CUSUM-UCB, while CU-LH is often slightly better: we believe that the reason why breakpoint prediction helps in this scenario is the fact that each change makes a sub-optimal arm still sub-optimal, thus the bias introduced by the past history still supports the idea that the arm is sub-optimal after the change.

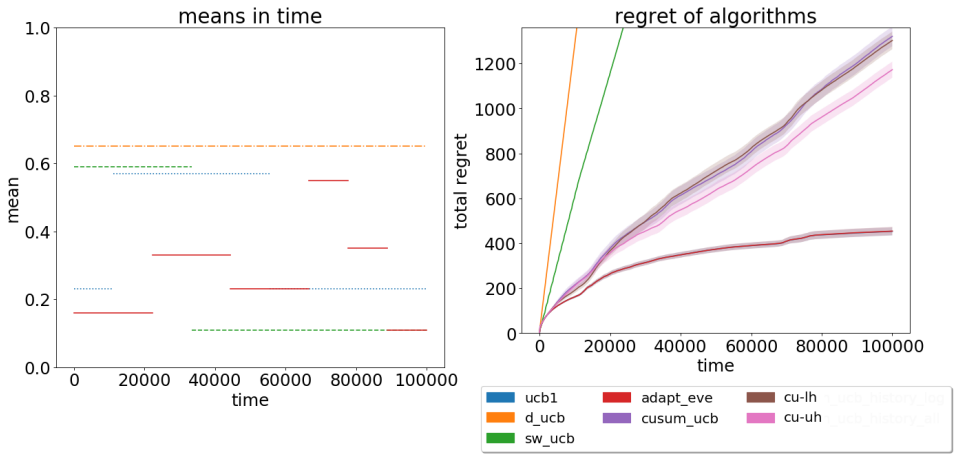


Figure 6.10: Scenario 3. Setting in which the optimal arm is stationary and the other arms are non-stationary, always having smaller mean than the stationary arm.

arms	2		8	
breakpoints	8	24	8	24
UCB1	**453** $\pm$ **18.4**	**335** $\pm$ **10.1**	**749** $\pm$ **11.4**	**599** $\pm$ **12.0**
D-UCB	14139 $\pm$ 12.6	14981 $\pm$ 9.4	23066 $\pm$ 11.6	27010 $\pm$ 8.3
SW-UCB	6349 $\pm$ 15.0	8981 $\pm$ 16.5	11840 $\pm$ 15.3	17211 $\pm$ 19.3
Adapt-EvE	**454** $\pm$ **18.5**	**336** $\pm$ **10.0**	**749** $\pm$ **11.4**	**599** $\pm$ **12.0**
CUSUM-UCB	1321 $\pm$ 43.6	1817 $\pm$ 43.0	2479 $\pm$ 69.0	3254 $\pm$ 72.2
CU-LH	1302 $\pm$ 39	1739 $\pm$ 35.5	2445 $\pm$ 78.1	3147 $\pm$ 65.3
CU-UH	1172 $\pm$ 35.6	1385 $\pm$ 29.6	2144 $\pm$ 67.4	2498 $\pm$ 35.9

Table 6.13: Scenario 3. 95% confidence bound on the pseudo-regret when there are 8 or 24 breakpoints.

6.2.4 Scenario 4: bounded changes

In this scenario, the magnitude of the changes on the expected reward is upper bounded and there is at least a breakpoint in which a suboptimal arm becomes the optimal one. We expect the bias introduced by past samples to be smaller than the one in the case of unbounded changes in Scenario 2, and, thus, we expect the variants of CUH to perform better, with respect to the results obtained in Scenario 2, than their counterpart without history CUSUM-UCB.

Results are shown in Tables 6.14, 6.15 and 6.16. Since, in the experiments, each arm starts with a random initial mean reward and increases and decreases in the mean occur with the same probability, it is likely that, since the changes are bounded, an arm that is optimal remains optimal across

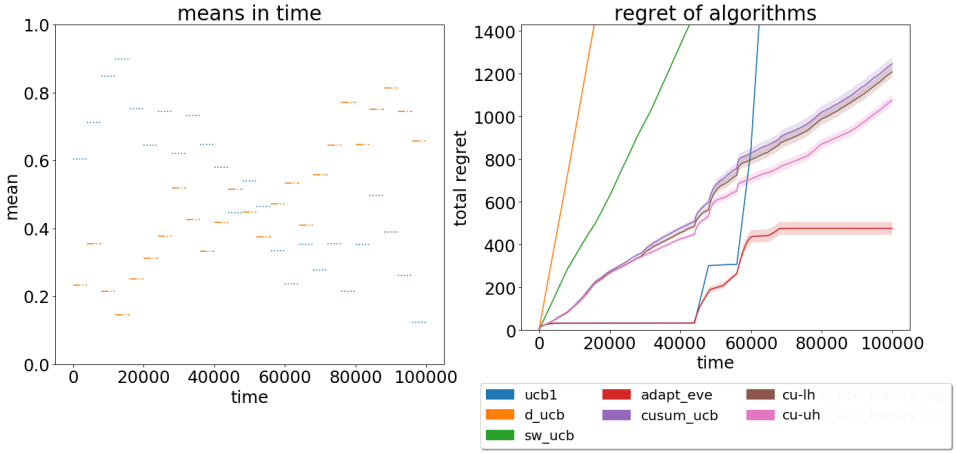


Figure 6.11: Scenario 4, example.

	2 arms			8 arms		
breakpoints	1	8	24	1	8	24
UCB1	504 ± 15	$\mathbf{995 \pm 28}$	1241 ± 16	$\mathbf{1752 \pm 23}$	3100 ± 32	$\mathbf{2613 \pm 37}$
D-UCB	2754 ± 6	17163 ± 14	4597 ± 7	22437 ± 10	5991 ± 8	24876 ± 8
SW-UCB	1753 ± 11	7751 ± 19	2582 ± 10	12865 ± 18	3273 ± 11	15968 ± 17
Adapt-EvE	$\mathbf{378 \pm 27}$	1002 ± 28	873 ± 25	$\mathbf{1765 \pm 36}$	1963 ± 65	$\mathbf{2613 \pm 57}$
CUSUM-UCB	561 ± 57	1613 ± 98	916 ± 50	2835 ± 103	1163 ± 39	4070 ± 96
CU-LH	557 ± 52	1561 ± 96	873 ± 46	2743 ± 103	1130 ± 40	3881 ± 84
CU-UH	461 ± 43	1376 ± 92	$\mathbf{703 \pm 39}$	2337 ± 92	$\mathbf{942 \pm 32}$	3388 ± 78

Table 6.14: Scenario 4. 95% confidence bound on the pseudo-regret when the maximum mean change is 0.1.

several breakpoints. Thus, many changes are like the ones in Scenario 3, i.e., sub-optimal arms remain sub-optimal and optimal arms remain optimal. In most experiments, the optimal arm is the same for most of the time, and so UCB1 achieves small regret, but, as soon as the optimal arm changes once, UCB1 achieves large regret, as shown in Figure 6.11. Like in Scenario 3, CU-UH always performs significantly better than CUSUM-UCB, while CU-LH has always a lower regret but never significantly, and we see that, the more past history is used, the lower is the regret.

	2 arms			8 arms		
breakpoints	1	8	24	1	8	24
UCB1	729 ± 17	1888 ± 22	981 ± 24	2641 ± 34	5613 ± 109	4611 ± 57
D-UCB	3507 ± 9	17141 ± 12	5714 ± 9	23986 ± 10	6027 ± 7	24983 ± 8
SW-UCB	1856 ± 13	8501 ± 20	2565 ± 10	13502 ± 21	3301 ± 11	15954 ± 16
Adapt-EvE	$\mathbf{435 \pm 35}$	1922 ± 22	919 ± 24	$\mathbf{2250 \pm 40}$	4118 ± 47	$\mathbf{3598 \pm 76}$
CUSUM-UCB	481 ± 38	1501 ± 73	931 ± 42	2818 ± 91	1178 ± 38	4591 ± 98
CU-LH	470 ± 37	1474 ± 72	878 ± 42	2761 ± 92	1137 ± 39	4415 ± 100
CU-UH	$\mathbf{391 \pm 30}$	$\mathbf{1195 \pm 51}$	$\mathbf{711 \pm 31}$	$\mathbf{2318 \pm 73}$	$\mathbf{960 \pm 32}$	3852 ± 94

Table 6.15: Scenario 4. 95% confidence bound on the pseudo-regret when the maximum mean change is 0.15.

	2 arms			8 arms		
breakpoints	1	8	24	1	8	24
UCB1	1398 ± 22	$\mathbf{1568 \pm 96}$	3486 ± 101	$\mathbf{2737 \pm 44}$	3500 ± 22	$\mathbf{4038 \pm 104}$
D-UCB	3522 ± 9	16530 ± 13	5386 ± 9	23847 ± 10	6056 ± 7	26663 ± 9
SW-UCB	1726 ± 11	8043 ± 20	2705 ± 11	13223 ± 20	3200 ± 10	16575 ± 17
Adapt-EvE	584 ± 75	$\mathbf{1577 \pm 93}$	2955 ± 141	$\mathbf{2779 \pm 52}$	2784 ± 53	$\mathbf{3865 \pm 94}$
CUSUM-UCB	$\mathbf{410 \pm 29}$	2319 ± 170	1057 ± 53	3312 ± 102	1439 ± 39	4652 ± 99
CU-LH	$\mathbf{409 \pm 28}$	2264 ± 167	1003 ± 46	3163 ± 102	1363 ± 36	4460 ± 88
CU-UH	$\mathbf{363 \pm 26}$	1987 ± 156	$\mathbf{846 \pm 42}$	$\mathbf{2723 \pm 101}$	$\mathbf{1189 \pm 35}$	$\mathbf{3977 \pm 94}$

Table 6.16: Scenario 4. 95% confidence bound on the pseudo-regret when the maximum mean change is 0.2.

6.2.5 Scenario 5: real-world data with Yahoo!

This scenario is about investigating the behaviour of the selected algorithms with real-world data. The chosen dataset is the Yahoo! dataset [42], which contains information on the click through rate of articles displayed in the Featured Tab of the Today Module on Yahoo!'s front page. The data is extracted as explained in Section 6.1. Overall results are organised in Table 6.17. In all the chosen days, CU-UH is significantly better than all the other policies, proving its strategy to be good also on real-life scenarios. CU-LH seems to achieve slightly lower regret than CUSUM-UCB, but not significantly. The usefulness of breakpoint prediction in this scenario can be explained by the fact that, in this dataset, changes often have small magnitudes and, thus, it is likely that a sub-optimal arm remains sub-optimal, as in Scenarios 3 and 4. UCB1 achieves lower regret than their passively adaptive counterparts D-UCB and SW-UCB because, despite the presence of breakpoints, they often do not involve a suboptimal arm becoming the

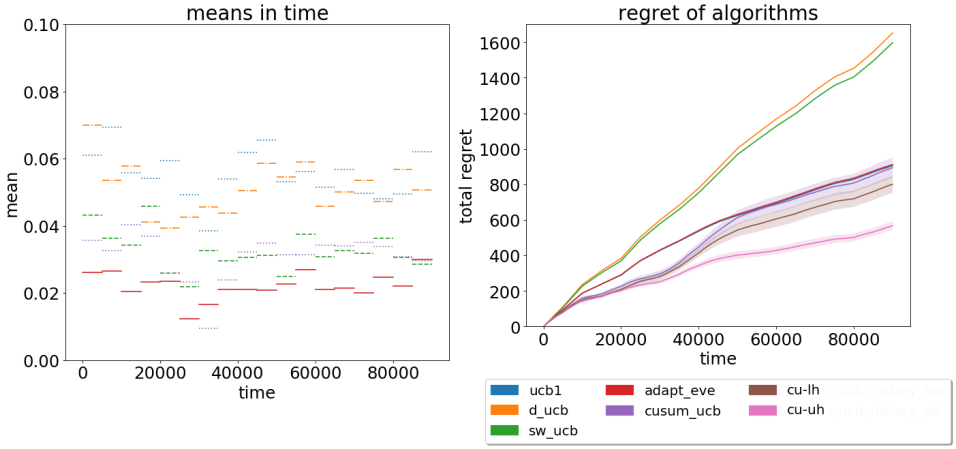


Figure 6.12: Scenario 5. Experiment on the first selected day.

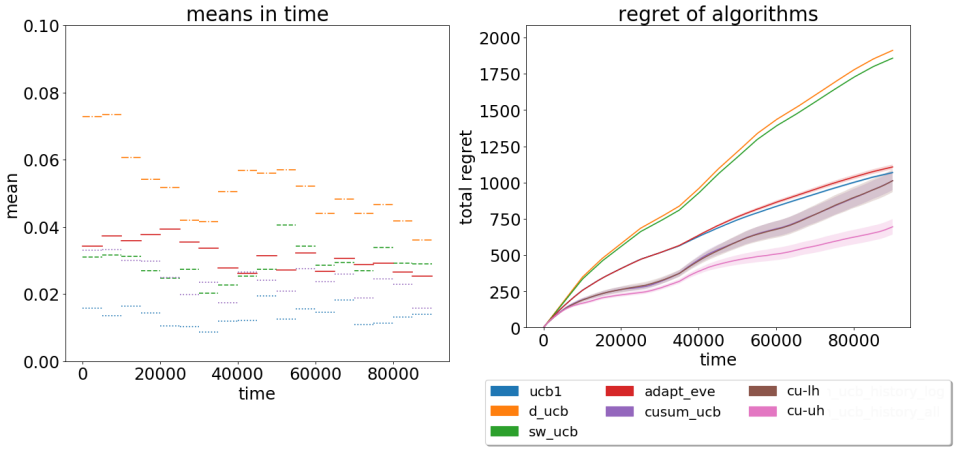


Figure 6.13: Scenario 5. Experiment on the second selected day.

optimal arm.

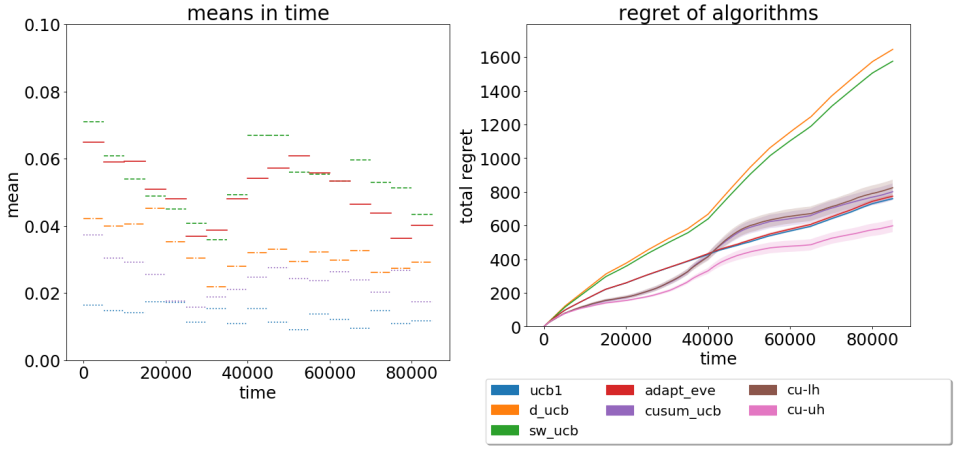


Figure 6.14: Scenario 5. Experiment on the third selected day.

	Day 1	Day 2	Day 3
UCB1	910 ± 6.1	1067 ± 7.5	759 ± 6.0
D-UCB	165 ± 0.36	1912 ± 0.3	1645 ± 0.4
SW-UCB	1598 ± 0.8	1858 ± 0.8	1575 ± 0.9
Adapt-EvE	912 ± 7.1	1118 ± 18.7	776 ± 12.5
CUSUM-UCB	791 ± 45.7	1031 ± 74.2	816 ± 44.2
CU-LH	793 ± 56.1	994 ± 70.7	782 ± 46.8
CUH-UH	$\mathbf{583 \pm 31.8}$	$\mathbf{669 \pm 58.5}$	$\mathbf{616 \pm 42.8}$

Table 6.17: Scenario 5. 95% confidence bound on the pseudo-regret obtained on the Yahoo! dataset.

Chapter 7

Conclusion

In this thesis, we analysed the Abruptly-Changing Non-Stationary MAB problem and the abvailable state-of-the-art policies, which can be subdivided in passively adaptive and actively adaptive. Encouraged by recent results, both theoretical and practical, on actively adaptive policies, we propose a novel algorithm, called CUH, with the goal of improving in both areas. In particular, our policy takes advantage of the breakpoint prediction capability of CDTs, already studied in statistical analysis but not yet employed by state-of-the-art multi-armed bandit algorithms.

Regarding the theoretical analysis, our contribution is two-fold: first, we prove a regret upper bound for an actively adaptive meta-policy which is independent from the CDT and the stationary subpolicy used, which can be either frequentist or bayesian, provided that an upper bound of the number of rounds required to gather $\mathbb{E}[D]$ samples on a suboptimal arm is specified. Then, we prove a regret upper bound for the CUH policy when the maximum amount of history is fixed, which is the first result, up to our knowledge, concerning a MAB policy using breakpoint prediction.

Regarding practical results, we test our novel policy and state-of-the-art algorithms with extensive experiments over different synthetic scenarios and a real-world dataset of webpage click-through rates. The outcomes demonstrate that CUH with unbounded history outperforms existing policies on the real-world dataset and on most synthetic scenarios, while CUH with bounded history performs similarly to its counterpart without history.

Encouraged by the practical results, future works can focus on analysing the theoretical performance of the CUH policy with unbounded history, and test the policy with other sub-policies different from UCB1-Explore, such as TS.

Acronyms

A-MAB Adversarial Multi-Armed Bandit. 4, 5, 9, 19, 20

Adapt-EvE Adaptive Exploration versus Exploration. 21, 23, 33, 36, 37, 39, 65–67, 70, 71, 74, 75, 79

ANS-MAB Abruptly changing Non-Stationary Multi-Armed Bandit. 5, 6, 20, 30, 31, 34, 36, 38, 43, 45, 51, 62, 65

CDT Changepoint Detection Test. 20, 22, 25, 30, 33, 34, 36–39, 41, 43, 45–48, 50–55, 61–63, 65, 71, 85

CU-LH CUSUM-UCB with Logarithmic History. 65, 67, 71–73, 76, 79, 81, 82

CU-UH CUSUM-UCB with Unbounded History. 65, 67, 71–73, 76, 79, 81, 82

CUH CUSUM-UCB with History. 2, 21, 43–47, 53, 54, 62, 65–67, 70–72, 74, 75, 79, 80, 85

CUSUM Cumulative Sum control chart. 20–23, 25, 34, 37, 38, 43, 45, 55, 56, 58, 61, 62, 66

CUSUM-UCB Cumulative Sum control chart UCB. 21, 34, 37, 39–41, 43, 44, 65–67, 70–76, 79–82

D-UCB Discounted UCB. 33, 34, 65, 82

Exp3 Exponential-weight algorithm for Exploration and Exploitation. 19, 20

KL-UCB Kullback Leibler Upper Confidence Bound. 14

MAB Multi-Armed Bandit. 1–6, 8, 10, 12–14, 19, 20, 29

MDP Markov Decision Process. 3, 4

NS-MAB Non-Stationary Multi-Armed Bandit. 5, 7, 22, 29–31, 33, 65

PHT Page-Hinkley Test. 21, 23, 36

S-MAB Stochastic Multi-Armed Bandit. 4

SNS-MAB Smoothly changing Non-Stationary Multi-Armed Bandit. 5, 6

ST-MAB Stationary Multi-Armed Bandit. 4, 8, 13, 14, 16, 18, 33, 43, 46, 51, 52, 61, 62

SW-UCB Sliding Window UCB. 33, 35, 36, 65, 82

TS Thompson Sampling. 8, 11, 15–19, 85

Two-Sided CUSUM Two-Sided Cumulative Sum control chart. 39, 43, 54

UCB1 Upper Confidence Bound 1. 8–10, 12–14, 33–35, 39, 41, 48, 65, 71, 72, 79, 81, 82

UCB1-Explore UCB1 with uniform Exploration. 34, 37, 39, 43, 45, 47, 51–56, 61, 62, 66, 85

UCB1-Tuned Upper Confidence Bound 1 Tuned. 10, 14, 33, 36

Bibliography

[1] William Robin Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3):285–294, 1933.

[2] Anisio Lacerda. Multi-objective ranked bandits for recommender systems. *Neurocomputing*, 246:12 – 24, 2017. Brazilian Conference on Intelligent Systems 2015.

[3] Eric M. Schwartz, Eric T. Bradlow, and Peter S. Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522, 2017.

[4] Y. Chen, Q. Zhao, V. Krishnamurthy, and D. Djonin. Transmission scheduling for optimizing sensor network lifetime: A stochastic shortest path approach. *IEEE Transactions on Signal Processing*, 55(5):2294–2309, May 2007.

[5] Cédric Hartland, Sylvain Gelly, Nicolas Baskiotis, Olivier Teytaud, and Michèle Sebag. Multi-armed bandit, dynamic environments and meta-bandits. Working paper, Nov 2006.

[6] Fang Liu, Joohyun Lee, and Ness B. Shroff. A change-detection based framework for piecewise-stationary multi-armed bandit problem. *CoRR*, abs/1711.03539, 2018.

[7] E. S. Page. On problems in which a change in a parameter occurs at an unknown point. *Biometrika*, 44(1/2):248–252, 1957.

[8] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, May 2002.

[9] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331, Oct 1995.

[10] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, pages 174–188, 2011.

[11] Luis Da Costa, Álvaro Fialho, Marc Schoenauer, and Michèle Sebag. Adaptive Operator Selection with Dynamic Multi-Armed Bandits. In *Genetic and Evolutionary Computation Conference (GECCO)*, GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation, pages 913–920, Atlanta, United States, July 2008. ACM.

[12] Weiwei Shen, Jun Wang, Yu-Gang Jiang, and Hongyuan Zha. Portfolio choices with orthogonal bandit learning. In *IJCAI*, 2015.

[13] A. Motamedi and A. Bahai. Mac protocol design for spectrum-agile wireless networks: Stochastic control approach. In *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 448–451, April 2007.

[14] Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits. In *COLT*, 2008.

[15] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 22, 1985.

[16] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[17] Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1):55–65, Sep 2010.

[18] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 661–670, New York, NY, USA, 2010. ACM.

[19] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 586–594. Curran Associates, Inc., 2010.

[20] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, May 1997.

[21] Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *J. Mach. Learn. Res.*, 11:2785–2836, December 2010.

[22] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *CoRR*, abs/1204.5721, 2012.

[23] Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. *CoRR*, abs/1402.6028, 2014.

[24] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial Intelligence and Statistics*, pages 99–107, 2013.

[25] Emilie Kaufmann, Olivier Cappe, and Aurelien Garivier. On bayesian upper confidence bounds for bandit problems. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 592–600, La Palma, Canary Islands, 21–23 Apr 2012. PMLR.

[26] Toshihiro Kamishima and Shotaro Akaho. Personalized pricing recommender system: Multi-stage epsilon-greedy approach. In *Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 57–64, New York, NY, USA, 2011. ACM.

[27] Joannès Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In João Gama, Rui Camacho, Pavel B. Brazdil, Alípio Mário Jorge, and Luís Torgo, editors, *Machine Learning: ECML 2005*, pages 437–448, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[28] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[29] Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *COLT*, 2011.

[30] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003.

[31] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Tuning bandit algorithms in stochastic environments. In Marcus Hutter, Rocco A.

Servedio, and Eiji Takimoto, editors, *Algorithmic Learning Theory*, pages 150–165, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[32] E. Weisstein. Beta distribution. From MathWorld—A Wolfram Web Resource.

[33] Junya Honda and Akimichi Takemura. Optimality of thompson sampling for gaussian bandits depends on priors. In *AISTATS*, 2014.

[34] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2249–2257. Curran Associates, Inc., 2011.

[35] D. Lu, P. Mausel, E. Brondízio, and E. Moran. Change detection techniques. *International Journal of Remote Sensing*, 25(12):2365–2401, 2004.

[36] Levente Kocsis and Csaba Szepesvári. Discounted ucb. video lecture, 2006.

[37] D Hinkley. Inference about the change-point from cumulative sum tests. *Biometrika*, 58, 12 1971.

[38] G. Lorden. Procedures for reacting to a change in distribution. *Ann. Math. Statist.*, 42(6):1897–1908, 12 1971.

[39] Michèle Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice Hall, Inc. - http://people.irisa.fr/Michele.Basseville/kniga/, 1993.

[40] Rasul A. Khan. A note on page's two-sided cumulative sum procedure. *Biometrika*, 68(3):717–719, 1981.

[41] E. Weisstein. Tree. From MathWorld—A Wolfram Web Resource.

[42] Yahoo! R6b - yahoo! front page today module user click log dataset, version 2.0, 2011. data retrieved from `https://webscope.sandbox.yahoo.com/catalog.php?datatype=r`.