Bandits multi-armés avec rétroaction partielle

Pratik Gajane

École Doctorale Sciences pour l'Ingénieur
Inria Lille - Nord Europe
Université Lille 3

THÈSE DE DOCTORAT

présentée pour obtenir le grade de
DOCTEUR EN SCIENCES DE L'UNIVERSITÉ LILLE 3
Spécialité : **Informatique**

présentée par
Pratik GAJANE

MULTI-ARMED BANDITS WITH UNCONVENTIONAL FEEDBACK

sous la direction de M Philippe **PREUX**
et le co-encadrement de M Tanguy **URVOY**

Rapporteurs:	M. Aurelien **GARIVIER**	Institut de Mathématiques de Toulouse
	M. Maarten **De RIJKE**	University of Amsterdam

Soutenue publiquement le 14 Novembre 2017 devant le jury composé de :

Alexandra **CARPENTIER**	Institut für Mathematik, Universität Potsdam	Examinatrice
Richard **COMBES**	Centrale-Supelec	Examinateur
Maarteen **De RIJKE**	University of Amsterdam	Rapporteur
Aurélien **GARIVIER**	Institut de Mathématiques de Toulouse	Rapporteur
Emilie **KAUFMANN**	CNRS, Université de Lille	Examinatrice
Gabor **LUGOSI**	Pompeu Fabra University	Président
Philippe **PREUX**	Université de Lille	Directeur
Tanguy **URVOY**	Orange Labs	Co-encadrant

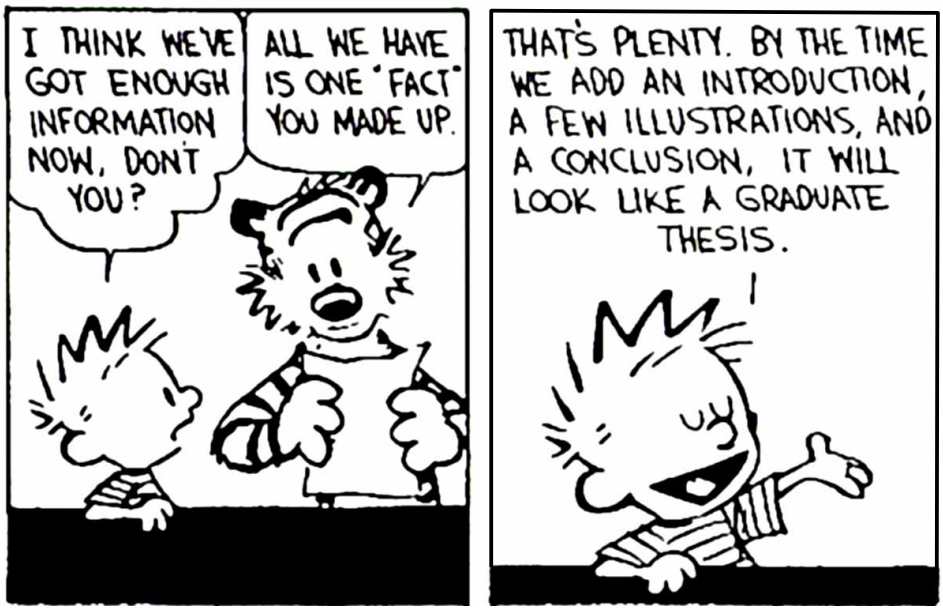
Declaration of Authorship

I, Pratik GAJANE, declare that this thesis titled, "Multi-armed bandits with unconventional feedback" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:


I THINK WE'VE GOT ENOUGH INFORMATION NOW, DON'T YOU?
ALL WE HAVE IS ONE "FACT" YOU MADE UP.
THAT'S PLENTY. BY THE TIME WE ADD AN INTRODUCTION, A FEW ILLUSTRATIONS, AND A CONCLUSION, IT WILL LOOK LIKE A GRADUATE THESIS.

École Doctorale Sciences pour l'Ingénieur
INRIA Lille - Nord Europe
Université Lille 3

Abstract

Faculty Name
Department or School Name

Doctor of Philosophy

Multi-armed bandits with unconventional feedback

by Pratik GAJANE

Acknowledgements

Contents

Declaration of Authorship iii

Abstract vii

Acknowledgements ix

I Introduction and Preliminaries 1

1 Introduction and Preliminaries 3
1.1 Sequential decision making . . . 3
1.2 Multi-armed bandits . . . 5
1.2.1 Formalization of the problem . . . 6
Stochastic rewards . . . 6
Adversarial rewards . . . 7
1.2.2 Performance measure: regret . . . 7
1.2.3 Pure exploration setting . . . 8
1.2.4 Exploration-exploitation setting . . . 9
1.2.5 Best arm identification . . . 9
Fixed-confidence setting (PAC setting) . . . 10
Fixed-budget setting . . . 10
1.3 Applications of the MAB problem . . . 11
1.3.1 Clinical trials . . . 11
1.3.2 Internet advertising . . . 12
1.3.3 Online recommendation . . . 12
1.4 Algorithms for the MAB problem . . . 12
1.4.1 Algorithms for best arm identification with fixed confidence . . . 13
1.4.2 Algorithms for exploration-exploitation . . . 14
1.5 Partial monitoring . . . 17
1.5.1 Examples of partial monitoring game . . . 18
1.5.2 Hierarchy of finite stochastic partial monitoring games . . . 19
1.5.3 Expressing MAB problems as partial monitoring game . . . 21

II Dueling Bandits 23

2 The Dueling Bandits problem 25
2.1 Motivation . . . 25
2.2 Formalization . . . 28
2.2.1 Stochastic dueling bandits . . . 28
2.2.2 Adversarial dueling bandits . . . 28
2.2.3 Utility-based formulation . . . 29
Stochastic utility-based dueling bandits . . . 29

Adversarial utility-based dueling bandits . . . 31
2.2.4 Preference-based formulation . . . 31
Stochastic preference-based formulation . . . 32
Adversarial preference-based formulation . . . 32
2.2.5 Relation between utilities and preferences . . . 33
2.3 Our contributions . . . 33
2.4 Related work . . . 34
2.4.1 Dueling Bandit Gradient Descent . . . 34
2.4.2 Interleaved Filtering . . . 36
2.4.3 Beat The Mean . . . 37
2.4.4 Sensitivity Analysis of VAriables for Generic Exploration (SAVAGE) 39
2.4.5 Relative upper confidence bound (RUCB) . . . 40
2.4.6 Relative confidence sampling (RCS) . . . 41
2.4.7 Merge relative upper confidence bound (MERGERUCB) . . . 42
2.4.8 Copeland confidence bound (CCB) . . . 43
2.4.9 Contextual dueling bandits . . . 44
2.4.10 Double Thompson sampling for dueling bandits . . . 45
2.4.11 SPARRING . . . 46

3 The Lower Bound **49**

4 The Algorithm and its Analysis **51**
4.1 Relative Exponential-weight Algorithm for Exploration and Exploitation (REX3) . . . 51
4.2 Illustration of REX3 on a toy problem . . . 53
4.3 Upper bound on the regret of REX3 . . . 55

5 Empirical Evaluation **61**
5.1 Empirical validation of Corollary 4.1 . . . 62
5.2 Interleave filtering experiments . . . 63

6 Dueling Bandits as Partial Monitoring games **69**
6.1 Formalization of dueling bandits as Partial Monitoring game . . . 69
6.2 Dueling bandits in the partial monitoring hierarchy . . . 71
6.3 Partial monitoring algorithms and their use for dueling bandits . . . 73

Appendix A Detailed Proof of Theorem 4.1 **75**
A.1 Proof of eq. (A.4) . . . 78
A.2 Proof of Lemma 4.1 . . . 78
A.3 Proof of eq. (4.5) and (A.8) . . . 79

III Corrupt Bandits **81**

7 The Corrupt Bandits problem **83**
7.1 Motivation . . . 83
7.2 Formalization . . . 86
7.2.1 Stochastic setting . . . 86
7.2.2 Adversarial setting . . . 88
7.2.3 Randomized response as a corrupt bandit problem . . . 89
7.3 Our contributions . . . 90
7.4 Related work . . . 90

7.4.1 Positive and unlabeled learning . . . 90
7.4.2 Learning in the presence of feature noise . . . 92
7.4.3 Learning in the presence of label noise . . . 93
7.4.4 Noise addition for data privacy . . . 95

8 The Lower Bounds **97**
8.1 Lower bound on the sample complexity for best arm identification . . 97
8.2 Lower bound on the cumulative regret for exploration-exploitation setting . . . 100

9 The Algorithms and their Analyses **105**
9.1 Algorithms for best arm identification . . . 105
9.1.1 Median elimination for corrupt bandits . . . 105
9.1.2 Exponential-gap elimination for corrupt bandits . . . 106
9.2 Algorithms for exploration-exploitation setting . . . 107
9.2.1 kl-UCB for MAB with corrupted feedback (kl-UCB-CF) . . . 108
9.2.2 UCB1 for MAB with corrupted feedback (UCB-CF) . . . 111
9.2.3 Thompson sampling for MAB with corrupted feedback (TS-CF) . . . 111
9.2.4 kl-UCB-CF and TS-CF for MAB with randomized response . . 114

10 Corrupt Bandits to enforce Differential Privacy **115**
10.1 Introduction to differential privacy . . . 115
10.1.1 Why randomization? . . . 117
10.1.2 Why not simply use cryptosystems? . . . 117
10.2 Differential privacy in bandits . . . 118
10.2.1 Motivation for differential privacy in bandits . . . 118
10.2.2 Previous work . . . 119
Differentially private UCB Sampling . . . 119
DP-UCB-BOUND and DP-UCB-INT . . . 120
10.3 Corrupt bandits for differential privacy . . . 122

11 Empirical Evaluation **127**
11.1 Comparison of dedicated corrupt bandit algorithms with WRAPPER . . 128
11.2 Comparison between kl-UCB-CF, UCB-CF and TS-CF . . . 130
11.2.1 Comparison over a period of time . . . 130
11.2.2 Comparison with varying corruption parameters . . . 132
11.2.3 Comparison with varying level of differential privacy . . . 133

12 Corrupt Bandits as Partial Monitoring game **137**

Appendix B Proof for Theorem 9.1 **139**

Appendix C Proof for Theorem 9.2 **143**

Appendix D Proof for Theorem 9.3 **147**

Appendix E Proof for Theorem 9.4 **153**

IV Final Remarks **161**

13 Summary and future work **163**

List of Figures

1.1 Sequential decision making with complete feedback 4
1.2 Bandit game with stochastic rewards 6
1.3 Bandit game with oblivious adversarial rewards 7
1.4 Bandit game with malicious adversarial rewards 7
1.5 Partial monitoring game 18

2.1 Results of querying "partial feedback" on two search engines 26
2.2 Interleaved ranking 27

4.1 REX3: weights at $t = 1$ 53
4.2 REX3: weights at $t = 2$ 54
4.3 REX3: weights at $t = 3$ 55

5.1 Empirical validation of Corollary 4.1. 62
5.2 Regret and accuracy plots averaged over 100 runs (50 runs for fixed-horizon algorithms) respectively on ARXIV dataset (6 rankers) and LETOR NP2004 dataset (64 rankers). 64
5.3 On the left: average regret and accuracy plots on MSLR30K with navigational queries ($K = 136$ rankers). On the right: same dataset, average regrets for a fixed $T = 10^5$ and K varying from 4 to 136. 64
5.4 Average regret and accuracy plots on LETOR NP2004 with respectively 16, and 32 rankers. 66
5.5 Expected regret and accuracy plots on MSLR30K with respectively informational and perfect navigational queries (136 rankers). 66
5.6 Average regret and accuracy plots respectively on LETOR NP2004 (64 rankers) and MSLR30K navigational queries (136 rankers) with Sparring coupled with a standard UCB MAB. 67
5.7 Average regret and accuracy plots respectively on SAVAGE and BVS artificial matrices. 67
5.8 An experiment on a synthetic utility-based 10-armed dueling bandits problem with non-stationary rewards. 68

6.1 Gain matrix $\mathcal{G}$ for a 4-armed binary dueling bandits resulting in 10 non-duplicate actions and 16 possible outcomes. 70
6.2 Feedback matrix $\mathcal{H}$ for the same problem as in Figure 6.1. 71
6.3 Signal matrix for action (12) for the same problem as in Figure 6.1. . . 71

8.1 In Figure 8.1a, g_a is such that $\lambda_a = g_a(\mu_1)$ thereby making it impossible to discern arm a from the optimal arm given the mean feedback. In Figure 8.1b, a steep monotonic g_a leads the reward gap $\Delta_a = \mu_1 - \mu_a$ into a clear gap between λ_a and $g_a(\mu_1)$. 103

10.1 Internet advertising with privacy preserving output 123
10.2 Internet advertising with privacy preserving input 124

11.1 Regret plots comparing dedicated corrupt bandit algorithms with WRAPPER . . . 129
11.2 Regret plots comparing dedicated corrupt bandit algorithms with WRAPPER . . . 129
11.3 Regret plots comparing dedicated corrupt bandit algorithms with WRAPPER . . . 129
11.4 Regret plots for comparison over a period of time . . . 131
11.5 Regret plots for comparison over a period of time . . . 131
11.6 Regret plots for comparison over a period of time . . . 131
11.7 Regret plots with varying corruption parameters . . . 132
11.8 Regret plots with varying corruption parameters . . . 133
11.9 Regret plots with varying corruption parameters . . . 133
11.10 Regret plots with varying level of differential privacy . . . 134
11.11 Regret plots with varying level of differential privacy . . . 134
11.12 Regret plots with varying level of differential privacy . . . 134

12.1 Gain matrix $\mathcal{G}$ and feedback matrix $\mathcal{H}$ for a 2-armed binary MAB problem with corrupted feedback resulting in 2 non-duplicate actions and 16 possible outcomes. . . . 138
12.2 Signal matrices for the same problem as in Figure 12.1. . . . 138

List of Tables

2.1 Summary of dueling bandit algorithms 47

5.1 The preference matrices used for experiments 62

6.1 Summary of partial monitoring algorithms 74

11.1 Bernoulli mean arm rewards for experimental scenarios. 128

For/Dedicated to/To my…

Part I

Introduction and Preliminaries

Chapter 1

Introduction and Preliminaries

The two major themes of this thesis are the *dueling bandits* and the *corrupt bandits* which are both variants of the multi-armed bandit (MAB) problem with unconventional forms of feedback. In this introductory chapter, we lay the foundation of the thesis by introducing the conventional MAB problem. We also introduce the more general concept of *partial monitoring*. We emphasize that the aim of this chapter is not to present an extensive and complete survey of these two vast fields but merely to introduce the key notions which aid the reading of the thesis. For a survey on multi-armed bandits, we point the readers to Bubeck and Cesa-Bianchi [2012]. For further reading on partial monitoring, we recommend Bartók et al. [2014].

This chapter is organized as follows: Firstly, we briefly describe the sequential decision making problem in Section 1.1, since the MAB problem is a sequential decision making problem with a form of incomplete feedback called *bandit feedback*, as we shall see in Section 1.2. We enlist in Section 1.3 some of the major practical applications of the MAB problem. In Section 1.4, we take a look at a few algorithms for the various settings of the MAB problem. As we are dealing with forms of feedback which differ from the conventional bandit feedback, it is natural to pose the considered problems in a more general paradigm for sequential decision making. This paradigm known as partial monitoring in introduced in Section 1.5.

1.1 Sequential decision making

As the name suggests, sequential decision making proceeds in a sequence of consecutive rounds. In each round, the learner has a number of available actions and its task is to select one to be taken. Action selection is based on the value associated

with each action by the environment. In the context of this thesis, a learner is simply a system which interacts with the environment and makes the decisions of selecting actions and the environment is anything external to the learner. At this point, we make no statistical assumptions about how the actions values are generated by the environment. The learner's goal is to select an action during each round to optimize the associated value. In order to do so, the learner forms suitable estimations for all the action values. Based on these estimations, the learner chooses one of the available actions accordingly. At the end of the round, feedback about the action value/s is revealed to the learner. Using this feedback, the learner can update the estimates for the action values in the next round. For a more detailed portrayal, please refer to Littman [1996, Chapter 1].

The most descriptive feedback that could be available to the learner is the observation of all the action values. We term such feedback as *complete feedback*. Sequential decision making with complete feedback is depicted in Figure 1.1. The key feature

Sequential decision making with complete feedback

At round (or time) $t = 1, \ldots,$

1. The environment assigns the values to all the available actions.
2. The learner chooses an action.
3. The learner receives the reward for the selected action.
4. The learner observers the rewards for all the actions.

FIGURE 1.1: Sequential decision making with complete feedback

of the complete feedback is the availability of all the action values at all the time periods, even for the actions not taken by the learner. While complete feedback is available in some scenarios like portfolio management, it is still a strong assumption. A natural relaxation of this assumption leads us to bandit feedback in which the learner, at any time period, only receives the feedback for the action taken and not for the remaining actions. Bandit feedback is motivated from practical applications like clinical trials, Internet advertising and online recommendation which are detailed in Section 1.3. A MAB problem is a mathematical formulation of sequential decision making with bandit feedback. In the next section, we formally define the

MAB problem and its various settings.

1.2 Multi-armed bandits

In a MAB problem, actions are symbolized by arms in reference to the arm of a slot machine or a one-armed bandit and selecting an action is symbolized by pulling the corresponding arm [1]. A learner has to iteratively pull an arm from a set of available arms. With each arm, there is an associated value. On selecting a particular arm, the learner receives the value (gains the reward or suffers the loss) corresponding to the arm it chose. For the purpose of this thesis, we work exclusively with rewards and not losses as they can be considered mirror images. As a feedback, the learner only observes the received reward corresponding to the selected arm and is given no other information as to the merit of other available arms. The learner's goal is to optimize the reward of the arms it chooses.

The learner can maintain the history of previous selections and the subsequent rewards it received and observed as feedback to estimate the action rewards for the next round. Higher the number of times a particular arm is selected, more accurate is the estimate of its reward. At any round, the learner can decide to select an arm with the current highest reward estimate. Such a choice is called a *greedy* choice. With a greedy choice, the learner is said to *exploit* its current knowledge of the action rewards. However, if the learner has inaccurate estimates of the other arm rewards then one of the those arms might turn out to have a higher reward than the greedy choice. Hence the learner can decide to choose an arm which currently has an inferior reward estimate in order to have a more accurate estimate of its reward. With such a non-greedy choice, the learner is said to *explore*. With exploration, received reward is inferior in the short run, but if the exploration leads to discovery of arms with rewards higher than that of the current greedy choice, then the learner can obtain superior reward in the longer run by exploiting the said better arms. Hence the learner faces the exploration-exploitation dilemma which is inherent to reinforcement learning problems. This dilemma is mathematically formalized as a MAB problem as follows.

[1] For the rest of thesis, we shall use pulling or selecting or choosing an arm interchangeably.

1.2.1 Formalization of the problem

It is symbolized as a repeated game between a learner and the environment. The learner has a set of arms $A = \{1, \ldots, K\}$ available to it. At every time period $t = 1, \ldots,$ each arm a is associated with a numerical reward. The reward vector $\mathbf{x}_t$ consists of $\{x_1(t), \ldots, x_K(t)\}$ where $x_a(t)$ [2] is the reward associated with the arm a at time t. Simultaneously at time period t, the learner pulls an arm a_t and receives the reward $x_{a_t}(t)$. Observation of the received reward $x_a(t)$ i.e. bandit feedback is available to the learner and it does not have access to the rewards of the other arms.

For most of the settings we consider, the game described above is restricted to a finite time period $\{1, \ldots, T\}$ where T is called the *horizon*. In *finite-horizon* setting the horizon is known to the learner and it is unknown to the learner in *anytime* setting.

For most of the problems, we assume the rewards are bounded in $[0, 1]$. In a binary multi-armed bandit or a Bernoulli multi-armed bandit problem, the reward values are restricted to 0 and 1. The reward values can be generated by the following two ways:

Stochastic rewards

The crux of this formulation is the presence of the stationary reward probability associated with each arm. There are K probability distributions $\nu_1, \ldots, \nu_K$ over $[0, 1]$ associated respectively with arms $1, \ldots, K$. Let $\mu_1, \ldots, \mu_K$ be the respective means of $\nu_1, \ldots, \nu_K$. When an arm a is pulled, its reward x_a is drawn from the corresponding distribution ν_a. This setting can be described as a game between the learner and the environment as follows:

Bandit game with stochastic rewards

At $t = 1, \ldots, T$

1. The environment draws a reward vector $\mathbf{x}_t$ according to $\nu_1 \times \cdots \times \nu_K$.
2. The learner chooses an arm $a_t \in A = \{1, \ldots, K\}$.
3. The learner observes and receives the reward $x_{a_t}(t)$.

FIGURE 1.2: Bandit game with stochastic rewards

[2] When t is clear from the context, we simply write x_a instead of $x_a(t)$.

Adversarial rewards

Unlike in the case of stochastic rewards, the reward probabilities may not be stationary but the rewards are generated by an adversary. The adversary can be either *oblivious* or *malicious*. An oblivious adversary selects all the reward vectors $\mathbf{x}_1, \ldots, \mathbf{x}_T$ beforehand. It can be described as a game given below.

Bandit game with oblivious adversarial rewards

The adversary draws reward vectors $\mathbf{x}_t \in [0,1]^K$ for $t = 1, \ldots, T$.
At $t = 1, \ldots, T$

1. The learner chooses an arm $a_t \in A = \{1, \ldots, K\}$.
2. The learner observes and receives the reward $x_{a_t}(t)$.

FIGURE 1.3: Bandit game with oblivious adversarial rewards

However, a malicious adversary chooses a reward vector $\mathbf{x}_t$ at time t having access to $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$ and $a_1, \ldots, a_{t-1}$. It can be described as a game below.

Bandit game with malicious adversarial rewards

At $t = 1, \ldots, T$

1. The adversary draws a reward vector $\mathbf{x}_t \in [0,1]^K$.
2. The learner chooses an arm $a_t \in A = \{1, \ldots, K\}$.
3. The learner observes and receives the reward $x_{a_t}(t)$.
4. The adversary observes the learner's arm selection a_t and the received reward $x_{a_t}(t)$.

FIGURE 1.4: Bandit game with malicious adversarial rewards

1.2.2 Performance measure: regret

A *policy* is a mapping from time period $t \in \{1, \ldots, T\}$ to $a_t \in A$ i.e. the arm selected at time t. Let the policy with the highest associated arm reward be called the optimal policy. The optimal policy could be single-armed or multiple-armed. If the learner knew the optimal policy beforehand, then at every time period, it would simply pull

the arm given by the optimal policy and receive the highest possible reward. However, in most problems worth solving, the learner does not have access to the optimal policy. The learner has to conjecture the best arm from the available bandit feedback i.e. the rewards of the previously chosen arms. In other words, the learner's goal is to find a policy which is the closest approximation to the optimal policy. Every time the learner pulls a sub-optimal arm , it is losing out on the difference of the rewards between those two arms. Since the learner's goal is to maximize the reward of its chosen arms, it follows that the learner should try to minimize this difference i.e. the regret.

Regret can be classified into two kinds depending upon whether the exploration and the exploitation overlap. Based on these two kinds of regret, which we will see shortly, the MAB problem can be divided into two settings: *pure exploration* and *exploration-exploitation*. The third MAB setting, called *best arm identification*, uses other performance measures. All of these three settings are introduced next.

1.2.3 Pure exploration setting

In this setting, introduced by Bubeck et al. [2009], the learner has to deal with two tasks. The secondary task is of exploration i.e of selecting an arm for sampling at every time period. Based on the rewards it observes for the selected arms, the learner's primary task is to select an arm at the end of each time period to be used as a recommendation if/when the environment sends a stopping signal indicating that the exploration phase is over. The goal of the learner is to minimize the *simple regret*.

Definition 1.1. ***Simple regret*** $(\mathrm{SRegret})$***:*** *Simple regret is defined as the difference between the expected reward of the optimal arm (in hindsight) and the expected reward of the arm recommended by the learner.*

In other words, the learner first explores through the arms in the exploration phase to gain knowledge about the rewards of the arms and then it exploits the acquired knowledge to recommend an arm. The exploration phase and the exploitation phase do not overlap. The rewards of the arms selected by the learner during the exploration phase are not considered for computing the simple regret; only the reward of the recommended arm is considered.

1.2.4 Exploration-exploitation setting

In this setting, introduced by Robbins [1952], the exploration phase and the exploitation phase overlap. The learner's task is not to recommend an arm at the end of the game, but to find a policy which selects an arm at every time period such that the *cumulative regret* is minimized.

Definition 1.2. ***Cumulative regret*** (CRegret)***:*** *Cumulative regret is defined as the difference between the expected cumulative reward of the optimal policy and the expected cumulative reward of the learner's policy.*

So while exploring through the arms, the learner also has to exploit its knowledge about the rewards of the arms as the rewards of the selected arms are being considered for the computation of the cumulative regret. This captures the classic exploration vs exploitation dilemma in reinforcement learning.

Cumulative regret can be computed against a single arm optimal policy, in which case it is called the *weak regret*, or a dynamic multi-arm optimal policy, in which case it is called the *strong regret*.

$$\text{Weak regret} = \mathbb{E}\left[\sum_{t=1}^{T} \max(\mathbf{x}_t)\right] - \sum_{t=1}^{T} x_{a_t}(t)$$

$$\text{Strong regret} = \sum_{t=1}^{T} \mathbb{E}(\max(\mathbf{x}_t)) - \sum_{t=1}^{T} x_{a_t}(t)$$

In all of the cases where this setting is used in this thesis, we use the notion of weak regret.

1.2.5 Best arm identification

The early occurrences of this setting are considered by Bechhofer [1958] and Paulson [1964]. In this setting, by convention, the rewards are assumed to be stochastic. The arm with highest mean reward is called the optimal arm or the best arm and is denoted by a_* i.e. $a_* := \underset{a=1,\ldots,K}{\operatorname{argmax}}\ \mu_a$. The corresponding optimal mean reward is denoted by μ_*.

Unlike the settings of pure exploration and exploration-exploitation, this setting does not use regret as the measure of performance. The learner, as usual, samples

the available arms to acquire knowledge about their rewards. The learner's goal is to recommend an approximation of the best arm. The learner's decision to stop sampling the arms and make its recommendation is influenced by either fixing the *confidence* in the recommendation or fixing the *budget* for sampling.

Fixed-confidence setting (PAC setting)

In fixed-confidence best arm identification setting, considered by both Bechhofer [1958] and Paulson [1964], the learner is obliged to recommend an arm with a certain level of confidence defined by two parameters ϵ and δ. The parameter ϵ is used to indicate the degree of acceptable approximation for the recommended arm, while δ is the maximum allowable error probability. The learner's goal is to recommend an ϵ-approximate arm, i.e. an arm a having the mean reward $\mu_a \geq \mu_* - \epsilon$, with the probability of at-least $1 - \delta$. This is also called as *Probably approximate correct* (PAC) setting. The performance of the learner is measured in terms of the *sample complexity* which is the number of samples required by the learner to achieve its goal.

Fixed-budget setting

In fixed-budget best arm identification setting, introduced by Audibert et al. [2010], every arm a is associated with a cost c_a known to the learner. The learner has to pay the cost c_a every time it decides to sample the corresponding arm a. The learner is further given a fixed budget which specifies the maximum permissible cumulative cost. The learner's goal is to recommend an ϵ-approximate arm before exhausting the given budget. The cost for sampling could be different for all the arms. This setting addresses the best possible use of available resources (e.g. cpu time) in order to optimize the performance of some decision-making task. That is, it is used to model situations with a preliminary exploration phase in which costs are measured in terms of resources constrained by a limited budget. In a special case, the cost for every arm could be set to 1 to restrict the total number of samples the learner is allowed to make. The probability with which the learner fails to a recommend an ϵ-approximate arm is termed as the error probability δ. The performance of the learner is measured in terms of the error probability.

1.3 Applications of the MAB problem

The original motivation of Thompson [1933] for studying the MAB problem came from clinical trials. Subsequently, the MAB problem has found its application in other fields as well. In this section we shall take a look at some practical applications of the MAB problem.

1.3.1 Clinical trials

In a clinical trial, a researcher would like to find the best treatment for a particular disease, out of many possible treatments. In a MAB formulation of a clinical trial, arms represent the treatments while sampling an arm signifies applying the corresponding treatment on the test subjects. We shall use this application to further explain the difference between the exploration-exploitation setting and best arm identification setting.

Consider a clinical trial for a severe disease in which a number of people suffering from the disease are used as the test subjects. In such a trial, the loss of trying a wrong treatment is high (or in terms of reward, the associated reward would equal a large negative value). It is important to minimize the cumulative regret, since the test and cure phases coincide. Therefore exploration-exploitation setting is suitable in this case.

On the other hand, consider a clinical trial for a cosmetic product in which various possible formulae are tested on animals used as the test subjects. In such a trial, the loss of trying a wrong formula is minimal, excusing the ethical concerns related to harming the animals. There exists a test phase in which all the formulae are tried on the test subjects without taking into consideration the incurred immediate loss. The test phase is limited by a fixed allocation of funds (fixed-budget) or the required level of quality of the recommendation (fixed-confidence). At the end of the test phase, the best-performing formula is recommended for the commercialization phase, and one aims at maximizing the quality of the recommended formula which is to be regarded as a commercialized product.

1.3.2 Internet advertising

Nowadays companies have a suite of potential online ads they can be displayed to the users, but they need to know which ad strategy to follow to maximize sales. This is similar to A/B testing, but has the added advantage of naturally minimizing strategies that do not work (and generalizes to A/B/C/D... strategies). A classical MAB problem can be utilised for this application but the presence of extra information or *context* paves the way for another setting of MAB problem called *contextual bandits*. Context is any additional information that can be used to make a better decision when choosing among all the ads. It includes user's age, location, previous buying habits, all of which can be highly informative of what type of products they might purchase in the future.

1.3.3 Online recommendation

On the Internet, a huge amount of digital information hinders the users from accessing the items they are interested in. To solve this problem, online recommender systems provide personalized item recommendations to users. Recommender systems are beneficial for online vendors too as they enhance their revenues by providing them effective means of showcasing the products the users are more likely to buy. The scenario of online recommendation can be modeled as a contextual MAB problem by considering the items as the arms to be selected by the learner i.e. the recommender system.

1.4 Algorithms for the MAB problem

In this section, we take a look at some of the popular algorithms for some of the variations of the MAB problem. These algorithms form the basis of the algorithms introduced in this thesis (Chapters 4 and 9). Firstly, we shall see two algorithms for best arm identification with fixed confidence.

1.4.1 Algorithms for best arm identification with fixed confidence

Many algorithms for the best arm identification with fixed confidence are elimination strategies that work in rounds. Elimination strategies can be described succinctly using the following:

1. Sampling rule: to decide which arm (or arms) to pull during a round.
2. Elimination rule: to decide which arm (or arms) to eliminate at the end of a round.
3. Stopping rule: to decide when to stop sampling and recommend an arm.

We provide below the earliest and the simplest elimination algorithm for best arm identification with fixed confidence.

- **Median elimination:** Median elimination (ME), given by Even-Dar et al. [2006], eliminates the worst half of the arms at each iteration. The step 4 of ME (given in Algorithm 1) specifies the sampling rule, in which every remaining arm is pulled a certain number of times depending upon the approximation parameter ϵ, the error probability δ and the current round number. The mean of the observed rewards for each arm serves as the reward estimate for the arm. By the elimination rule, given in step 6, the arms with lower reward estimate than the median reward estimate are eliminated. The algorithm stops when there is only one arm remaining and recommends the remaining arm. This algorithm serves as a baseline for the algorithm ME-CF, which we introduce in Section 9.1.1.

Algorithm 1 Median elimination (ME)

Input: A bandit model with a set of arms $A := \{1, \ldots, K\}$ arms with unknown reward means $\mu_1, \ldots, \mu_K$.

1: **Parameters:** $\epsilon > 0$, $\delta > 0$
2: Set $S_1 \leftarrow A, \epsilon_1 \leftarrow \epsilon/4, \delta \leftarrow \delta/2, l \leftarrow 1$
3: **Do**
4: Sample every arm a in S_l for $1/(\epsilon_l/2)^2 \log(3/\delta_l)$ times and let $\hat{\mu}_a^l$ be its mean empirical reward.
5: Let Med_l be the median of $\{\hat{\mu}_a^l\}_{a \in S}$.
6: $S_{l+1} \leftarrow S_l \setminus \{a : \hat{\mu}_a^l < \mathrm{Med}_l\}$
7: $\epsilon_{l+1} \leftarrow \frac{3}{4}\epsilon_l, \delta_{l+1} \leftarrow \frac{\delta}{2}, l \leftarrow l + 1$
8: **Until** $|S_l| = 1$
9: Output the arm in S_l

- **Exponential gap elimination:** Exponential gap elimination (EGE), given by Karnin et al. [2013], aims to eliminate $(1/2)^l$-suboptimal arms at round l. This algorithm uses ME as a subroutine to estimate the suboptimality of each arm. EGE is described in Algorithm 2. This algorithm serves as a baseline for the algorithm EGE-CF, which we introduce in Section 9.1.2.

Algorithm 2 Exponential-gap elimination (EGE)

Input: A bandit model with a set of arms $A \coloneqq \{1, \ldots, K\}$ arms with unknown reward means $\mu_1, \ldots, \mu_K$ $\frac{1}{\sigma}$.
Parameters: $\delta > 0$
Set $S_1 \leftarrow A, l \leftarrow 1$
While $|S_l| > 1$
    let $\epsilon_l \leftarrow 2^{-l}/4$ and $\delta_l \leftarrow \delta/(50l^3)$
    Sample each arm $a \in S_l$ for $(2/\epsilon_l^2)\log(2/\delta_l)$ and let $\hat{\mu}_a^l$ be its mean empirical reward.
    Invoke $a_l \leftarrow \text{ME}(S_l, \epsilon_l/2, \delta_l)$
    Set $S_{l+1} \leftarrow S_l \setminus \{a \in S_l : \hat{\mu}_a^l < \hat{\mu}_{a_l}^l - \epsilon_l\}$
    $l \leftarrow l + 1$
End while
Output the arm in S_l

1.4.2 Algorithms for exploration-exploitation

- **UCB1:** Upper confidence bound policies are classical algorithms for the stochastic MAB problem. They work on the principle of "optimism in the face of uncertainty". These policies compute an *upper confidence bound* (UCB) for each arm and use it as the estimate of its reward. At each time period, the arm with the highest upper confidence bound is pulled. UCB1 (Algorithm 3), given by Auer et al. [2002a], is the simplest UCB algorithm. This algorithm serves a baseline for the algorithm UCB-CF, which we introduce in Section 9.2.2.

Algorithm 3 UCB1

Input: A bandit model with a set of arms $A \coloneqq \{1, \ldots, K\}$ arms with unknown reward means $\mu_1, \ldots, \mu_K$.
Initialization: Play each arm once.
for $t = K + 1, \ldots$ **do**
    Pull arm $\hat{a}_t = \text{argmax}_a \hat{x}_a + \sqrt{\frac{2 \cdot \log K}{N_a(t-1)}}$, where $\hat{x}_a$ is the average reward from arm a and $N_a(t-1)$ is the number of arms pulled till time $t-1$.
end for

- kl-**UCB:** kl-UCB, given by Cappé et al. [2013], is an upper confidence bound based policy for the stochastic MAB problem. It uses upper confidence bounds for the arm rewards based on Kullback-Leibler divergence. The precise description of kl-UCB is given in Algorithm 4. The empirical mean of the reward obtained from the arm a until time t is denoted by $\hat{\mu}_a(t)$. This algorithm serves a baseline for the algorithm kl-UCB-CF, which we introduce in Section 9.2.1.

Algorithm 4 kl-UCB

Input: A bandit model with a set of arms $A \coloneqq \{1, \ldots, K\}$ arms with unknown mean rewards $\mu_1, \ldots, \mu_K$.

Parameters: A non-decreasing (exploration) function $f : \mathbb{N} \to \mathbb{R}$, $d(x, y) \coloneqq \mathrm{KL}(\mathcal{B}(x), \mathcal{B}(y))$, Time horizon T.

1: **Initialization:** Pull each arm once.
2: **for** time $t = K, \ldots, T-1$ **do**
3: Compute for each arm a in A the quantity

$$\mathrm{Index}_a(t) \coloneqq \sup\{q : \ N_a(t) \cdot d(\hat{\mu}_a(t), q) \leq f(t)\}$$

4: Pull arm $\hat{a}_{t+1} \coloneqq \underset{a \in A}{\mathrm{argmax}} \ \mathrm{Index}_a(t)$.
5: **end for**

- **Exponential-weight algorithm for Exploration and Exploitation (EXP3):** EXP3, given by Auer et al. [2002b], is a randomized algorithm for the adversarial MAB problem. It is a variant of the Hedge algorithm introduced by Freund and Schapire [1997].

  At every time step t, EXP3 pulls an arm $\hat{a}_t$ according to a distribution which is a mixture of the uniform distribution and a distribution which assigns to each arm a probability mass exponential in the estimated reward for that action. Therefore the algorithm, at all times, selects every arm with at-least a non-zero probability of γ/K, thus ensuring continual exploration, where γ is the exploration parameter and K is the number of arms. For computing the estimated reward, the algorithm makes use of importance sampling. This choice guarantees that the expectation of the estimated reward for each arm is equal to its

Algorithm 5 EXP3

Input: A bandit model with a set $A = \{1, \dots, K\}$
Parameters: Real $\gamma = (0, 1]$
Initialization: $w_a(1) = 1$ for $a \in A$
for $t = 1, 2, \dots$ **do**
  Set
$$p_a(t) = (1 - \gamma)\frac{w_a(t)}{\sum_{b=1}^{K} w_b(t)} + \frac{\gamma}{K} \qquad a = 1, \dots, K$$
  Draw $\hat{a}_t$ randomly according to the probabilities $p_1(t), \dots, p_K(t)$
  Observe the receive $x_{\hat{a}_t}(t) \in [0, 1]$
  for $b = 1, \dots, k$ **do**, set
$$\hat{x}_b(t) = \begin{cases} x_b(t)/p_b(t) & \text{if } b = a_t, \\ 0 & \text{otherwise} \end{cases}$$
  end for
end for

actual reward. This algorithm serves a baseline for the algorithm REX3, which we introduce in Section 4.1.

- **Thompson sampling:** Thompson sampling (TS), given by Thompson [1933], is an algorithm for the stochastic MAB problem. It follows a Bayesian approach, where Bayesian priors are used as a tool to encode the current knowledge about the arm rewards. TS maintains a Beta posterior distribution on the

Algorithm 6 Thompson sampling for Bernoulli bandits

Input: A Bernoulli bandit model with a set of arms $A \coloneqq \{1, \dots, K\}$ arms with unknown reward means $\mu_1, \dots, \mu_K$.
Initialization: For each arm a in A, set $\text{success}_a = 0$ and $\text{fail}_a = 0$
for $t = 0, \dots$ **do**.
  For each arm a in A, sample $\theta_a(t)$ from $\text{Beta}(\text{success}_a + 1, \text{fail}_a + 1)$
  Pull arm $\hat{a}_{t+1} \coloneqq \arg\max_a \theta_a(t)$ and receive the reward $x_{\hat{a}_t}(t + 1)$
  if $x_{\hat{a}_t}(t + 1) = 1$ **then**
    $\text{success}_{\hat{a}_{t+1}} = \text{success}_{\hat{a}_{t+1}} + 1$
  else
    $\text{fail}_{\hat{a}_{t+1}} = \text{fail}_{\hat{a}_{t+1}} + 1$
  end if
end for

mean reward of each arm. At round $t+1$, for each arm a, it draws a sample $\theta_a(t)$ from the posterior distribution on coreeposnding to arm a and pulls the arm for which $g_a^{-1}(\theta_a(t))$ is largest. This mechanism ensures that at each round, the probability that arm a is played is the posterior probability of this arm to be optimal. Algorithm 6 describes the Thompson sampling algorithm for stochastic

bandits with Bernoulli rewards. Agrawal and Goyal [2012] provide a simple extension which works for bandits with arbitrary reward distributions with support $[0, 1]$. This algorithm serves a baseline for the algorithm TS-CF, which we introduce in Section 9.2.3.

In the next section, we take a look at a general paradigm for sequential decision making with incomplete feedback.

1.5 Partial monitoring

Partial Monitoring (PM) provides a generic mathematical model for sequential decision making with incomplete feedback. In this section, we take a brief review of the basic concepts of partial monitoring problems. Most of the information in this section is taken from Bartók et al. [2011] and Bartók [2013].

A partial monitoring game (PM) is defined by a tuple $\langle \boldsymbol{N}, \boldsymbol{M}, \boldsymbol{\Sigma}, \mathcal{L}, \mathcal{H} \rangle$ where $\boldsymbol{N}$, $\boldsymbol{M}$, $\boldsymbol{\Sigma}$, $\mathcal{L}$ and $\mathcal{H}$ are the action set, the outcome set, the feedback alphabet, the loss function and the feedback function respectively. To each action $I \in \boldsymbol{N}$ and outcome $J \in \boldsymbol{M}$, the loss function $\mathcal{L}$ associates a real-valued loss $\mathcal{L}(I, J)$ and the feedback function $\mathcal{H}$ associates a feedback symbol $\mathcal{H}(I, J) \in \boldsymbol{\Sigma}$.

In every round, the opponent and the learner simultaneously choose an outcome J_t from $\boldsymbol{M}$ and an action I_t from $\boldsymbol{N}$, respectively. The learner then suffers the loss $\mathcal{L}(I_t, J_t)$ and receives the feedback $\mathcal{H}(I_t, J_t)$. Only the feedback is revealed to the learner, the outcome and the reward remain hidden. In some problems, gain $\mathcal{G}$ is considered instead of loss. The loss function $\mathcal{L}$ and the feedback function $\mathcal{H}$ are known to the learner. When both $\boldsymbol{N}$ and $\boldsymbol{M}$ are finite, the reward function and the feedback function can be encoded by matrices, namely reward matrix and feedback matrix each of size $|\boldsymbol{N}| \times |\boldsymbol{M}|$. We take the liberty of overloading the notations $\mathcal{L}$ and $\mathcal{H}$ to also mean loss matrix and feedback matrix respectively. The learner's aim is to control the cumulative regret against the best single-action policy at time T:

$$\mathrm{CRegret}_T = \max_i \sum_{t=1}^{T} \mathcal{L}(I_t, J_t) - \mathcal{L}(i, J_t)$$

Partial monitoring game

At time $t = 1, \ldots,$

1. The opponent chooses an outcome $J_t \in \boldsymbol{M}$.
2. The learner chooses an action $I_t \in \boldsymbol{N}$.
3. The learner suffers the loss $\mathcal{L}(I_t, J_t)$ (or receives the the reward $\mathcal{G}(I_t, J_t)$).
4. The learner observers the feedback $\mathcal{H}(I_t, J_t) \in \boldsymbol{\Sigma}$.

FIGURE 1.5: Partial monitoring game

1.5.1 Examples of partial monitoring game

Various interesting problems can be modeled as partial monitoring games, such as the multi-armed bandit problem, learning with expect advice (Littlestone and Warmuth [1994]), dynamic pricing (Kleinberg and Leighton [2003]), the dark pool problem (Agarwal et al. [2010]), label efficient prediction (Cesa-bianchi et al. [2005]), and linear and convex optimization with full or bandit feedback (Zinkevich [2003a], Abernethy et al. [2008], Flaxman et al. [2004]).

The Bernoulli multi-armed bandit problem: A partial monitoring formulation of this problem is provided with a set of K arms/actions $i \in \boldsymbol{N} = \{1, \ldots, K\}$, an alphabet $\boldsymbol{\Sigma} = [0, 1]$, and a set of environment outcomes which are vectors $\boldsymbol{m} \in \boldsymbol{M} = [0, 1]^K$. The entry with index i ($\boldsymbol{m}_i$) denotes the instantaneous gain of the i^{th} arm. Assuming binary gains, $\boldsymbol{M}$ is finite and of size 2^K.

$$\mathcal{G}(i, \boldsymbol{m}) = \boldsymbol{m}_i \qquad \mathcal{H}(i, \boldsymbol{m}) = \boldsymbol{m}_i$$

The dynamic pricing problem: A seller has a product to sell and the customers wish to buy it. At each time period, the customer secretly decides on a maximum amount she is willing to pay and the seller sets a selling price. If the selling price is below the maximum amount the buyer is willing to pay, she buys the product and the seller's gain is the selling price she fixed. If the selling price is too expensive, her gain is zero. The feedback is incomplete because the seller only receives binary information stating whether the customer has bought the product or not. A PM

formulation of this problem is provided below:

$$x \in \boldsymbol{N} \subseteq \mathbb{R}, \quad y \in \boldsymbol{M} \subseteq \mathbb{R}, \quad \boldsymbol{\Sigma} = \{\text{“sold", “not sold"}\}$$

$$\mathcal{G}(x,y) = \begin{cases} 0, & \text{if } x > y, \\ x, & \text{if } x \leq y, \end{cases} \qquad \mathcal{H}(x,y) = \begin{cases} \text{“not sold"}, & \text{if } x > y, \\ \text{“sold"}, & \text{if } x \leq y, \end{cases}$$

1.5.2 Hierarchy of finite stochastic partial monitoring games

Consider a finite partial monitoring game with action set $\boldsymbol{N}$, outcome set $\boldsymbol{M}$, loss matrix $\mathcal{L}$ and feedback matrix $\mathcal{H}$. For any action $i \in \boldsymbol{N}$, loss vector $\boldsymbol{loss}_i$ denotes the column vector consisting of i^{th} row in $\mathcal{L}$. Correspondingly, gain vector $\boldsymbol{gain}_i$ denotes the column vector consisting of i^{th} row in $\mathcal{G}$. Let $\Delta_{|\boldsymbol{M}|}$ be the $|\boldsymbol{M}|-1$-dimensional probability simplex i.e. $\Delta_{|\boldsymbol{M}|} = \left\{\boldsymbol{q} \in [0,1]^{|\boldsymbol{M}|} \mid ||\boldsymbol{q}||_1 = 1\right\}$. For any outcome sequence of length T, the vector $\boldsymbol{q}$ denoting the relative frequencies with which each outcome occurs is in $\Delta_{|\boldsymbol{M}|}$. The cumulative loss of action i for this outcome sequence can hence be described as follows:

$$\sum_{t=1}^{T} \mathcal{L}(i, J_t) = T \cdot \boldsymbol{l}_i^\intercal \boldsymbol{q}$$

The vectors denoting the outcome frequencies can be thought of as the opponent strategies. These opponent strategies determine which action is optimal i.e. the action with the lowest cumulative loss. This induces a *cell decomposition* on $\Delta_{|\boldsymbol{M}|}$.

Definition 1.3 (Cells). *The cell of an action i is defined as*

$$C_i = \left\{\boldsymbol{q} \in \Delta_{|\boldsymbol{M}|} \mid \boldsymbol{l}_i^\intercal \boldsymbol{q} = \min_{j \in \boldsymbol{M}} \boldsymbol{l}_j^\intercal \boldsymbol{q}\right\}$$

In other words, a cell of an action consists of those opponent strategies in the probability simplex for which it is the optimal action. An action i is said to be *Pareto-optimal* if there exists an opponent strategy $\boldsymbol{q}$ such that the action i is optimal under $\boldsymbol{q}$. The actions whose cells have a positive $(|\boldsymbol{M}|-1)$-dimensional volume

are called *Strongly Pareto-optimal*. Actions that are Pareto-optimal but not strongly Pareto-optimal are called *degenerate*.

Definition 1.4 (Cell decomposition)**.** *The cells of strongly Pareto-optimal actions form a finite cover of Δ_M called as the cell-decomposition.*

Two actions cells i and j from the cell decomposition are *neighbors* if their intersection is an $(|\boldsymbol{M}| - 2)$-dimensional polytope. The actions corresponding to these cells are also called as *neighbors*. The raw feedback matrices can be 'standardized' by encoding their symbols in *signal matrices*:

Definition 1.5 (Signal matrices)**.** *For an action i, let $\sigma_1, \ldots, \sigma_{s_i} \in \Sigma$ be the symbols occurring in row i of $\mathcal{H}$. The signal matrix $\mathcal{S}_i$ of action i is defined as the incidence matrix of symbols and outcomes i.e. $\mathcal{S}_i(k, m) = \mathbb{1}_{\mathcal{H}(i,m)=\sigma_k} \quad k = 1, \ldots, s_i, \quad$ for $m \in \boldsymbol{M}$.*

Observability is a key notion to assess the difficulty of a PM problem in terms of regret $\mathrm{CRegret}_T$ against best action at time T.

Definition 1.6 (Observability)**.** *For actions i and j, we say that $\boldsymbol{l}_i - \boldsymbol{l}_j$ is* globally observable *if $\boldsymbol{l}_i - \boldsymbol{l}_j \in \operatorname{Im} \mathcal{S}^\intercal$. Where the global signal matrix $\mathcal{S}$ is obtained by stacking all signal matrices. Furthermore, if i and j are neighboring actions, then $\boldsymbol{l}_i - \boldsymbol{l}_j$ is called* locally observable *if $\boldsymbol{l}_i - \boldsymbol{l}_j \in \operatorname{Im} \mathcal{S}_{i,j}^\intercal$ where the local signal matrix $\mathcal{S}_{i,j}$ is obtained by stacking the signal matrices of all neighboring actions for i, j: $\mathcal{S}_k$ for $k \in \{k \in \boldsymbol{N} \mid C_i \cap C_j \subseteq C_k\}$.*

Theorem 1.1 (Classification of partial monitoring problems)**.** *Let $(\boldsymbol{N}, \boldsymbol{M}, \boldsymbol{\Sigma}, \mathcal{L}, \mathcal{H})$ be a partial monitoring game. Let $\{C_1, \ldots, C_k\}$ be its cell decomposition, with corresponding loss vectors $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_k$. The game falls into the following four regret categories.*

- $\mathrm{CRegret}_T = 0$ *if there exists an action with $C_i = \Delta_{|\boldsymbol{M}|}$. This case is called* trivial.
- $\mathrm{CRegret}_T \in \Theta(T)$ *if there exist two strongly Pareto-optimal actions i and j such that $\boldsymbol{l}_i - \boldsymbol{l}_j$ is not globally observable. This case is called* hopeless.
- $\mathrm{CRegret}_T \in \tilde{\Theta}(\sqrt{T})$ *if it is not trivial and for all pairs of (strongly Pareto-optimal) neighboring actions i and j, $\boldsymbol{l}_i - \boldsymbol{l}_j$ is locally observable. This case is called* easy.
- $\mathrm{CRegret}_T \in \Theta(T^{2/3})$ *if $\mathscr{G}$ is not hopeless and there exists a pair of neighboring actions i and j such that $\boldsymbol{l}_i - \boldsymbol{l}_j$ is not locally observable. This case is called* hard.

1.5.3 Expressing MAB problems as partial monitoring game

In Section 1.5.1 we described how a classical MAB problem can be expressed as a partial monitoring game. The formulation of a K-armed Bernoulli MAB problem as a partial monitoring game requires matrices of dimension $K \times 2^K$ for gain matrix and feedback matrix. Even for moderate values of K, this requirement is impractical.

In Part II of this thesis, we shall focus on the dueling bandit problem which is a MAB problem with unconventional feedback. In Chapter 6, we illustrate how the Bernoulli dueling bandits can be formulated as a partial monitoring game. This formulation too proves to be impractical due the large sizes of the gain matrix and the feedback matrix. In the same chapter, we see how the performance guarantees given by the general partial monitoring algorithms are not as tight as that of the algorithm we propose for the dueling bandit problem.

In Part III of this thesis, we introduce the corrupt bandit problem which is another MAB problem with unconventional feedback. In Chapter 12, we illustrate how the Bernoulli corrupt bandits can be formulated as a partial monitoring game. This formulation is also unsuitable due the large sizes of the gain matrix and the feedback matrix.

With this, we conclude the introductory part of the thesis. In the next two parts, we study in detail the MAB problem with different kinds of unconventional feedback.

Part II

Dueling Bandits

Chapter 2

The Dueling Bandits problem

In this part of the thesis, we consider the multi-armed bandit problem with a particular kind of unconventional feedback called relative feedback. In the classical multi-armed bandit problem, the learner receives absolute feedback about its choices. However as we shall see shortly, only relative feedback is available in many practical scenarios. This chapter provides an introduction to the MAB problem with relative feedback.

In Section 2.1, we motivate the relative feedback from the practical applications of the MAB problem. In Section 2.2, we formally define the MAB problem with relative feedback and its various settings. Our contributions to this problem are enlisted in Section 2.3. At the end, in Section 2.4, we take an overview of the related work.

2.1 Motivation

Humans find it much easier to choose from one of the given options rather than giving absolute feedback about only one choice. For example, if the following two questions are asked to a group of people :

1 : Which sport do you prefer - football or basketball?

2 : How do you rate football out of 50?

The first question would receive greater number of responses than the second question. Hence relative feedback is naturally suited to many practical applications

where humans are expected to provide feedback like user-perceived product preferences, where a relative perception: *"A is better than B"* is easier to obtain than its absolute counterpart: *"A's value is 42, B is worth 33"*.

A more commercial application of relative feedback comes from information retrieval systems where users provide *implicit feedback* about the provided results. This implicit feedback is collected in various ways e.g. a click on a link, a tap, or any monitored action of the user. In all these ways however, this kind of feedback is often strongly biased by the model itself (the user cannot click on a link which was not proposed).

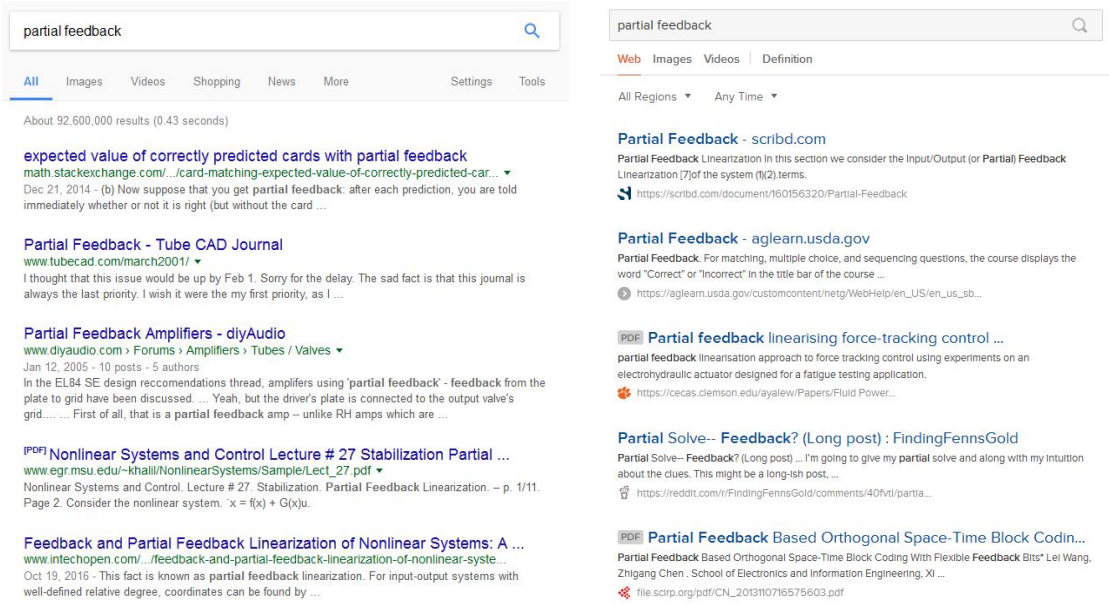


FIGURE 2.1: Results of querying "partial feedback" on two search engines

Consider the task of training search engines through machine learning. The figure 2.1 depicts a scenario where the same search query generates two varying results. While the user preference for either the first or the second result depends majorly upon whether their motivation behind the search has been satisfied, it is also affected by other factors such as presence of certain results and their rank on the page. In the past, click-through logs that passively collect user interactions with search engines were used as the source of training data. However Radlinski and Joachims [2007] showed that passively collecting data leads to the learned ranking never converging to an optimal ranking. This is because a number of studies have

shown that users tend to click on results ranked highly on search engines more often than the those ranked lower. Agichtein et al. [2006] study the click frequency on search engine results for 120,000 searches for 3,500 queries. They show that the relative number of clicks drop rapidly with the rank from the following observation in their study - compared to the top ranked result, 60% as many clicks on the second result, 50% as many clicks on the third, and 30% as many clicks on the fourth. This observation might lead to an interpretation that top ranked results are clicked more, simply because they are better. However Joachims et al. [2007] showed that users still click more often on higher ranked results even if the top ten results are presented in reverse order. In general, users very rarely see beyond the first page, so highly relevant search results that are not initially highly ranked may never be seen and evaluated.

To remove the aforementioned bias in search engines, Radlinski and Joachims [2007] propose to interleave the outputs of different ranking models. If the user clicks on the link from a certain ranking, that ranking is said to win the *duel*. The accuracy of this *interleaved ranking method* (figure 2.2) was highlighted in several experimental studies Radlinski and Joachims [2007], Joachims et al. [2007], Chapelle et al. [2012].

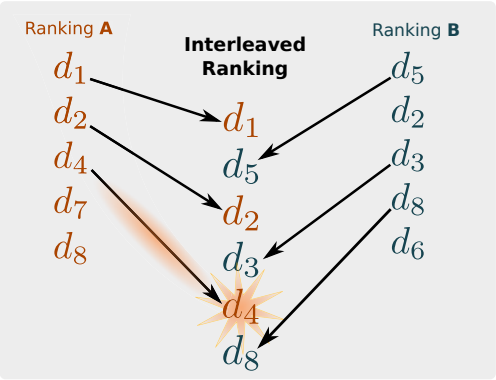


FIGURE 2.2: Interleaved ranking

Learning from relative feedback is relevant to many other fields, such as recommender systems (Gemmis et al. [2009]), and natural language processing (Callison-Burch et al. [2011]), which involve explicit or implicit feedback provided by humans. The classical MAB problem can not learn for such relative feedback provided in above scenarios. Hence it is crucial to devise a model which is able to learn from

relative feedback[1], which is presented in the next section.

2.2 Formalization

The *K-armed dueling bandit problem* is a variation of the classical Multi-Armed Bandit (MAB) problem introduced by Yue and Joachims [2009] to formalize the exploration/exploitation dilemma in learning from preference feedback. To be able to model the practical scenarios described in the previous section, the learner is to select two arms from the set $A = \{1, \ldots, K\}$ at every time period. As a feedback, the learner sees the information about which arm won the duel i.e. which arm gave higher reward. Note that the learner has no access to the rewards of the selected arms, but only to the preference feedback. However, the performance of the learner is judged on the basis of the rewards of the selected arms and not the feedback. The difficulty of this problem stems from the fact that the learner has no way of directly observing the reward of the selected arms. Hence the learner has to devise a way to infer from the feedback the necessary information about the rewards. This can be considered as an example of partial monitoring problem introduced in section 1.5. We discuss this further in Chapter 6.

Like the classical MAB problem (section 1.2.1), the dueling bandit problem too can be sub-divided based on the stationarity of the rewards into the following two categories:

2.2.1 Stochastic dueling bandits

Stochastic dueling bandit problem is characterized by stationarity like its analogue in the classical bandits described in section 1.2.1. However as for the dueling bandits, the learner sees only the relative feedback of the two selected arms and not their reward, the stationarity is exhibited by the preference of one arm over another arm.

2.2.2 Adversarial dueling bandits

In this formulation, the preference of one arm over another arm is non-stationary. At each time period, the adversary (or the environment) can affect the preference of

[1]The terms "relative feedback" and "preference feedback" are used interchangeably in this thesis.

every arm over every other arm.

In both the above categories, the preference of an arm over itself is assumed to be equal to 0.5. This is called into use when the arms are selected with replacement and an arm is selected with itself for a duel.

The dueling bandit problem can be formulated by either using utility values for individual arms or preference values for every pair of arms. Both of these formulations are described below.

2.2.3 Utility-based formulation

In this formulation, every arm is assigned a value. At each time period t, a real-valued bounded utility $x_a(t)$ is associated with each arm $a \in A$. When arms a and b are selected, their utility values determines which arm wins the duel as follows:

$$x_a(t) > x_b(t) : a \text{ wins the duel}$$
$$x_a(t) = x_b(t) : \text{tie}$$
$$x_a(t) < x_b(t) : b \text{ wins the duel}$$

A tie can be provided as a separate feedback or it can be broken randomly.

Now we describe how both the stochastic and adversarial dueling bandit problem can be encoded in utility-based formulation.

Stochastic utility-based dueling bandits

Every arm a is associated with a probability distribution ν_a with mean μ_a. When an arm a is selected by the learner, its utility is drawn from ν_a. The arm with the highest mean reward is called as the (an, in case of more than one) optimal arm a_*.

$$a_* = \underset{a \in A}{\operatorname{argmax}}\, \mu_a$$

The corresponding highest mean reward is denoted as μ_*. This formulation is depicted below:

Utilities drawn from $\nu_1, \ldots, \nu_K$	$x_1(t)$	$\ldots$	$x_a(t)$	$\ldots$	$x_b(t)$	$\ldots$	$x_K(t)$
			$\downarrow$		$\downarrow$		
The learner selects			a		b		
The learner sees			$\psi(x_a, x_b)$				
The learner receives			$\phi(x_a, x_b)$				

The feedback seen by the learner provides it relative information about the selected arms. One of the ways to provide such a relative feedback is as follows:

$$\psi(x_a, x_b) = \begin{cases} 1 & \text{if } x_a > x_b \\ 0 & \text{if } x_a = x_b \\ -1 & \text{if } x_a < x_b \end{cases}$$

On pulling arms a and b, the learner receives a reward given by $\phi(x_a, x_b)$. Depending upon the intent behind formulating the given dueling bandit problem, ϕ can take various forms. One way could be that the learner receives the highest of the two rewards i.e.

$$\phi(x_a, x_b) = \begin{cases} x_a & \text{if } x_a \geq x_b \\ x_b & \text{otherwise} \end{cases}$$

Another way is for the learner to receive the mean reward of the selected arms. In such cases, ϕ is as follows

$$\phi(x_a, x_b) = \frac{x_a + x_b}{2} \tag{2.1}$$

Using these notions, the cumulative reward can be defined as

$$\text{CRegret}_T = \sum_{t=1}^{T} \frac{2x_{a_*}(t) - x_a(t) - x_b(t)}{2} \tag{2.2}$$

where a_t and b_t are the two arms selected at time t till horizon T. In the rest of this thesis, we shall call this notion of regret as *bandit regret*.

Adversarial utility-based dueling bandits

At each time period, an adversary sets a bounded utility value for each of the arms. This formulation is depicted below:

The adversary sets	$x_1(t)$	$\dots$	$x_a(t)$	$\dots$	$x_b(t)$	$\dots$	$x_K(t)$
			$\downarrow$		$\downarrow$		
The learner selects			a		b		
The learner sees			$\psi(x_a, x_b)$				
The learner receives			$\phi(x_a, x_b)$				

2.2.4 Preference-based formulation

As the name suggests, this formulation is characterized by preferences. It is more expressive than utility-based formulation since it allows for a cycle. For example, the following preference can be expressed in this formulation unlike the utility-based formulation.

$$a \succ b \succ c \succ a$$

Since the preference of every pair of arms is to be specified, a natural way to represent all the preference values is by a square matrix with its dimension being equal to the number of arms when the number of arms are finite. It is called a *preference matrix* P. For each pair (a, b) of arms, the element $P_{a,b}$ contains the unknown probability with which a wins the duel against b.

An example preference matrix is shown below:

$$\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ K \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & K \end{array} \\ \begin{bmatrix} 1/2 & P_{1,2} & & P_{1,K} \\ P_{2,1} & 1/2 & & P_{2,K} \\ & & \ddots & \\ P_{K,1} & P_{K,2} & & 1/2 \end{bmatrix} \end{array}$$

The matrix P must satisfy the following symmetry property:

$$\forall a, b \in \{1, \ldots, K\}, \quad P_{a,b} + P_{b,a} = 1$$

Hence on the diagonal: $P_{a,a} = \frac{1}{2} \quad \forall a \in \{1, \ldots, K\}$.

Using these preference values, cumulative regret can be defined as

$$\text{CRegret}_T = \sum_{t=1}^{T} \frac{P_{a_*,a_t} + P_{a_*,b_t} - 1}{2} \tag{2.3}$$

where a_t and b_t are the two arms pulled at time t. In the rest of this chapter, this is called *Condorcet regret* since it coincides with the notion of a *Condorcet winner* a_*. A Condorcet winner is the arm, denoted by a_* which is preferred over all the other arms i.e. $\forall\, a \in A \setminus \{a_*\}, P_{a_*,a} > 1/2$. Note that if $\mu_a > \mu_b$ for some arms a and b, then $P_{a,b} > 1/2$. The optimal arm in the utility-based formulation thus coincides with the Condorcet winner in the preference formulation.

Stochastic preference-based formulation

As explained in section 2.2.1, the key characteristic of stochastic dueling bandits is the stationarity of the preference of one arm over another arm. The preference matrix directly stores this preference as a probability of arm a winning the duel against arm b for all a, b. Therefore the matrix based formulation lends itself easily to the stochastic dueling bandits.

Adversarial preference-based formulation

At each time period t, the adversary chooses a ternary square outcome matrix $\psi(t)$ where [2]

$$\psi(t)_{a,b} = \begin{cases} 1 & a \text{ wins the duel against } b \text{ at time } t \\ 0 & \text{a tie between } a \text{ and } b \text{ at time } t \\ -1 & b \text{ wins the duel against } a \text{ at time } t \end{cases} \tag{2.4}$$

[2] We overload the previously defined symbol for feedback function, ψ, to express a similar notion here.

Naturally the dimension of the outcome matrix is equal to the number of arms. At the horizon T, the preference matrix can be constructed as follows:

$$P_{a,b} = \frac{1}{2T} \sum_{t=1}^{T} \psi(t)_{a,b} + \frac{1}{2} \quad \forall a, b \in A$$

2.2.5 Relation between utilities and preferences

We can construct, preferences from utilities with randomized tie-breaking as follows:

$$P_{a,b} = \mathbb{P}(x_a > x_b) + \frac{1}{2}\mathbb{P}(x_a = x_b)$$

where $\mathbb{P}(E)$ indicates the probability of event E. When all v_a are Bernoulli laws, this reduces to:

$$P_{a,b} = \frac{\mu_a - \mu_b + 1}{2} \tag{2.5}$$

In this case, the bandit regret, given by Eq. (2.2), is twice the Condorcet regret, given by Eq. (2.3) as:

$$\frac{2\mu_{a_*} - \mu_a - \mu_b}{2} = P_{a_*,a} + P_{a_*,b} - 1$$

As utility-based formulation can not express cycles in preferences, there can be no such general way to convert preferences to utilities.

2.3 Our contributions

Our main contribution is an algorithm designed for the adversarial utility-based dueling bandit problem in contrast to most of the existing algorithms which assume a stochastic environment.

Our algorithm, called *Relative Exponential-weight algorithm for Exploration and Exploitation* (REX3), is a non-trivial extension of the *Exponential-weight algorithm for Exploration and Exploitation* (EXP3) algorithm Auer et al. [2002b] to the dueling bandit problem. We prove a finite time expected regret upper bound of order $\mathcal{O}(\sqrt{K \ln(K) T})$ and develop an argument initially proposed by Ailon et al. [2014] to exhibit a general lower bound of order $\Omega(\sqrt{KT})$ for this problem.

These two bounds correspond to the original bounds of the classical EXP3 algorithm and the upper bound strictly improves from the $\tilde{\mathcal{O}}(K\sqrt{T})$ obtained by existing generic partial monitoring algorithms.[3]

Our experiments on information retrieval datasets show that the anytime version of REX3 is a highly competitive algorithm for dueling bandits, especially in the initial phases of the runs where it clearly outperforms the state of the art.

We study the utility-based dueling bandit problem as an instance of the partial monitoring problem and prove that it fits the time-regret partial monitoring hierarchy as an *easy* – i.e. $\tilde{\Theta}\left(\sqrt{T}\right)$ – instance. We survey some partial monitoring algorithms and see how they could be used to solve dueling bandits efficiently.

2.4 Related work

The conventional MAB problem has been well studied in the stochastic setting as well as the (oblivious) adversarial setting (see section 1.2). The dueling bandits problem is recent, although related to previous works on computing with noisy comparison [see for instance Karp and Kleinberg, 2007]. This problem also falls under the framework of *preference learning* Freund et al. [2003], Liu [2009], Fürnkranz and Hüllermeier [2010] which deals with learning of (predictive) preference models from observed (or extracted) preference information i.e. relative feedback which specifies which of the chosen alternatives is preferred. Most of the articles hitherto published on dueling bandits consider the problem under a stochastic assumption.

2.4.1 Dueling Bandit Gradient Descent

Yue and Joachims [2009] consider the setting of stochastic utility-based dueling bandits with a possibly infinite number of actions. This is closely related to stochastic approximation (Robbins and Monro [1951]). The authors propose an algorithm called *Dueling Bandit Gradient Descent* (DBGD) to solve this problem. They approach this (*contextual*) dueling bandits problem with on-line convex optimization as follows: The set of actions A is embedded within a vector space W. Retrieval functions

[3] The notation $\tilde{\mathcal{O}}(\cdot)$ hides logarithmic factors.

in a search engine is an example of such a space. A is assumed to contain the origin, is compact, convex and is contained in a d-dimensional ball of radius r. They assume the existence of a differentiable, strictly concave utility function $v : A \to \mathbb{R}$. This function reflects the intrinsic quality of each point in A, and is never directly observed. Since v is strictly concave, there exists a unique maximum $v(a_*)$ where a_* is the optimum point (or action). A link function $\varphi : \mathbb{R} \to [0, 1]$ provides the following relation:

$$\forall a, b \in A : \mathbb{P}(a \text{ wins against } b) = \varphi(v(a) - v(b))$$

The function φ is assumed to be monotonic increasing, rotation-symmetric with a single inflection point at $\varphi(0) = 1/2$. In the search example, $\mathbb{P}(a \succ b)$ refers to the fraction of users who prefer the results provided by a over those of b. If at time t, the algorithm pulls arms a_t and b_t then its regret at time T is given by $\sum_{t=1}^{T} \varphi(v(a_*) - v(a_t)) + \varphi(v(a_*) - v(b_t)) - 1$.

Algorithm 7 Dueling Bandit Gradient Descent (DBGD)

Input: s_1, s_2, $a_1 \in W$
for t=1,...,T **do**
  Sample unit vector $\boldsymbol{u}_t$ uniformly.
  $b_t \leftarrow \text{project}_A(a_t + s_2 \boldsymbol{u}_t)$ *//projected back into A*
  Compare a_t and b_t
  if b_t wins **then**
    $a_{t+1} \leftarrow \text{project}_A(a_t + s_1 \boldsymbol{u}_t)$ *//projected back into A*
  else
    $a_{t+1} \leftarrow a_t$
  end if
end for

DBGD requires two parameters s_1 and s_2 which can be interpreted as exploitation and exploration step sizes respectively. The algorithm maintains a candidate a_t and compares it with a neighboring point b_t which is s_2 away from a_t. If b_t wins the comparison, an update of size s_1 is made along u_t, and then projected back into A(denoted by project_A). The authors prove that, if φ is L_1-Lipschitz and v is L_2-Lipschitz, and for suitable s_1 and s_2, the regret of DBGD is in $O(T^{3/4}\sqrt{rdL_1L_2})$ where r is the radius of the d-dimensional ball that is assumed to contain the set of actions A.

2.4.2 Interleaved Filtering

Yue et al. [2012] consider stochastic preference-based dueling bandit formulation where the preference matrix is expected to satisfy the assumptions of a strict linear order, strong stochastic transitivity and stochastic triangular inequality. They propose an algorithm called *Interleaved Filtering* (IF) for K-armed dueling bandit problem.

Algorithm 8 Interleaved Filtering (IF)

Input: T, $A = \{1, \ldots, K\}$.
$\delta \leftarrow 1/(TK^2)$
Choose an arm $\hat{a} \in A$ randomly
$W \leftarrow \{1, \ldots, k\} \setminus \{\hat{a}\}$
$\forall a \in W$, maintain estimate $\hat{P}_{\hat{a},a}$ of $P_{\hat{a},a}$
$\forall a \in W$, compute $1-\delta$ confidence interval $\hat{C}_{\hat{a},a} = (\hat{P}_{\hat{a},a} - c_t, \hat{P}_{\hat{a},a} + c_t)$ where $c_t = \sqrt{\log(1/\delta)/t}$
while $W \neq \varnothing$ **do**
    for $a \in W$ **do**
        compare a and $\hat{a}$
        update $\hat{P}_{\hat{a},a}$ and $\hat{C}_{\hat{a},a}$
    end for
    while $\exists a \in W$ s.t. $(\hat{P}_{\hat{a},a} > 1/2 \wedge 1/2 \notin \hat{C}_{\hat{a},a})$ **do**
        $W \leftarrow W \setminus \{a\}$ *//$\hat{a}$ declared winner against a*
    end while
    if $\exists b \in W$ s.t. $(\hat{P}_{\hat{a},b} < 1/2 \wedge 1/2 \notin \hat{C}_{\hat{a},b})$ **then**
        while $\exists b' \in W$ s.t. $\hat{P}_{\hat{a},b'} > 1/2$ **do**
            $W \leftarrow W \setminus \{b'\}$ *//pruning*
        end while
        $\hat{a} \leftarrow b$, $W \leftarrow W \setminus \{b\}$ *//b declared winner against $\hat{a}$ (new round)*
        $\forall b' \in W$, reset $\hat{P}_{\hat{a},b'}$ and $\hat{C}_{\hat{a},b'}$
    end if
end while
$\hat{T} \leftarrow$ Total comparisons made
return $(\hat{a}, \hat{T})$

Let a_* be the best arm. If (a_t, b_t) are the two arms selected at time t, strong regret for dueling bandits is defined as,

$$\text{StrongRegret} = \sum_{t=1}^{T} \max\{P_{a_*,a_t} - 1/2, P_{a_*,b_t} - 1/2\}$$

where T is the time horizon. Weak regret for dueling bandits is defined as,

$$\text{WeakRegret} = \sum_{t=1}^{T} \min\{P_{a_*,a_t} - 1/2, P_{a_*,b_t} - 1/2\}$$

Strong regret is computed by comparing the best arm to the worse of the pair of selected arms, while weak regret is computed by comparing the best arm to the better of the two selected arms. [4]

The algorithm IF is guaranteed to suffer an expected cumulative regret of order $\mathcal{O}(K \log T)$. This holds true for any notion of regret that is a linear combination of weak and strong regret as defined earlier.

2.4.3 Beat The Mean

Yue and Joachims [2011] consider the stochastic preference-based dueling bandits formulation where the preference matrix must only adhere to relaxed stochastic transitivity instead of strong stochastic transitivity necessitated by IF. They introduce *Beat The Mean* (BTM), an algorithm which proceeds by successive elimination of arms.

Algorithm 9 Beat-The-Mean (BTM)

Input: $A = \{1, \dots, K\}$, N, T, $c_{\delta,\gamma}(\cdot)$
$W_1 \leftarrow \{1, \dots, K\}$ *//working set of active arms*
$l \leftarrow 1$ (*num round*), $\forall a \in W_l, n_a \leftarrow 0$(*num comparisons*), $w_a \leftarrow 0$(*num wins*), $\hat{p}_a \equiv w_a/n_a$, or $1/2$ if $n_a = 0$
$n^* \coloneqq \min_{a \in W_l} n_a$, $c^* \coloneqq c_{\delta,\gamma}(n^*)$, or 1 if $n^* = 0$ *//confidence radius*
$t \leftarrow 0$ *//num iterations*
while $|W_l| > 1$ **and** $t < T$ **and** $n^* < N$ **do**
  $a \leftarrow \operatorname{argmin}_{a \in W_l} n_a$ *//break ties randomly*
  select $b \in W_l$ at random, compare a vs b
  If a wins, $w_a \leftarrow w_a + 1$
  $n_a \leftarrow n_a + 1$
  $t \leftarrow t + 1$
  if $\min_{a' \in W_l} \hat{p}_{a'} + c^* \leq \max_{a \in W_l} \hat{p}_a - c^*$ **then**
    $a' \leftarrow \operatorname{argmin}_{a' \in W_l} \hat{p}_{a'}$
    $\forall a \in W_l$, delete comparisons with a' from w_a, n_a
    $W_{l+1} \leftarrow W_l \setminus \{a'\}$ *//update working set*
    $l \leftarrow l + 1$ *//new round*
  end if
end while
return $\operatorname{argmax}_{a \in W_l} \hat{p}_a$

The algorithm proceeds in rounds and in each round l, a set of active arms is maintained as W_l. For each arm $a \in W_l$, an empirical estimate $\hat{p}_a$ is maintained. In each round, the the arm with the lowest number of comparisons is compared

[4] This distinction between strong and week regret is different than the one considered in Section 1.2

with the the arm b sampled uniformly from W_l, in effect comparing a with the *mean arm*. Whenever the worst empirical arm is separated from the best empirical arm by a sufficient confidence margin, the worst arm is eliminated from the working set along with all the comparisons involving it. The algorithm continues till W_l contains a single arm.

The authors prove that BTM (Algorithm 11) correctly returns the best arm with probability at least $1 - 1/T$ while accumulating the cumulative regret $\mathrm{CRegret} = O\left(\gamma^7 K \log T\right)$. This algorithm can be used for both exploration-exploitation setting as well as PAC setting.

- **Exploration-exploitation setting:** In the exploration-exploitation setting, the goal is to minimize cumulative Condorcet regret. In order to achieve minimum expected regret, the authors have adopted "explore then exploit" approach. In

Algorithm 10 Beat-The-Mean (Exploration-exploitation)

Input: $A = \{1, \ldots, K\}$, γ, T
1: $\delta \leftarrow 1/(2TK)$
2: $c_{\delta,\gamma}(n) := 3\gamma^2\sqrt{\frac{1}{n}\log\frac{1}{\delta}}$
3: Output Beat-The-Mean(A, ∞, T, $c_{\delta,\gamma}$)

the explore phase, BTM is called with the set of arms A, the horizon T, and the with the confidence interval $c_{\delta,\gamma}(n) = 3\gamma^2\sqrt{\frac{1}{n}\log(2KT)}$. In the exploit phase, the arm returned by the above invocation of BTM is pulled till horizon T. Thus the expected cumulative regret is bounded as follows:

$$\mathbb{E}[\mathrm{CRegret}_T] \leq (1 - 1/T)O\left(\gamma^7 K \log T\right) + (1/T)O(T) = O\left(\gamma^7 K \log T\right)$$

- **PAC setting:** In the PAC setting, the goal is to find an approximately optimal arm with a high probability using the minimum number of comparisons. The

Algorithm 11 Beat-The-Mean (PAC)

Input: $A = \{1, \ldots, K\}$, γ, ϵ, δ
1: Declare N to be the smallest integer such that $N = \left\lceil \frac{36\gamma^6}{\epsilon^2}\log\frac{K^3N}{\delta} \right\rceil$
2: $c_{\delta,\gamma}(n) := 3\gamma^2\sqrt{\frac{1}{n}\log\frac{K^3N}{\delta}}$
3: Output Beat-The-Mean(A, N, ∞, $c_{\delta,\gamma}$)

authors prove that BTM is an (ϵ, δ)-PAC algorithm with sample complexity

$$O\left(\frac{K\gamma^6}{\epsilon^2}\log\frac{KN}{\delta}\right)$$

where the confidence interval $c_{\delta,\gamma}(n) = 3\gamma^2\sqrt{\frac{1}{n}\log\frac{K^3N}{\delta}}$ and N is the smallest integer such that $N = \left\lceil\frac{36\gamma^6}{\epsilon^2}\log\frac{K^3N}{\delta}\right\rceil$.

2.4.4 Sensitivity Analysis of VAriables for Generic Exploration (SAVAGE)

Urvoy et al. [2013a] propose a generic algorithm called SAVAGE (for Sensitivity Analysis of VAriables for Generic Exploration) for stochastic preference-based dueling bandits. Their setting does away with several assumptions made in the previous algorithms e.g. existence of utility values or a linear order for the arms. In this general setting, the SAVAGE algorithm obtains a cumulative Condorcet regret bound of order $\mathcal{O}(K^2\log T)$. The key notions they introduce for dueling bandits are the *Copeland*, *Borda* and *Condorcet* scores (Charon and Hudry [2010]). The Borda score of an arm a on a preference matrix P is $\sum_{b=1}^{K} P_{a,b}$ and its Copeland score is $\sum_{b=1}^{K}\mathbb{1}_{\left(P_{a,b}>\frac{1}{2}\right)}$ (we use $\mathbb{1}$ to denote the indicator function). If an arm has a Copeland score of $K-1$, which means that it defeats all the other arms in the long run and it is a Condorcet winner. There exists however some datasets like MSLR30K [2012] where this Condorcet condition is not satisfied. It is however possible to define a robust *Copeland regret* which applies for any preference matrix.

Algorithm 12 SAVAGE

Input: $A = \{1, \ldots, K\}$, f, F, T, δ
$W \coloneqq \{1, \ldots, K\}, H \coloneqq F, s \coloneqq 1$
$\forall a \in W : \hat{\mu}_a \coloneqq 1/2$ and $t_a \coloneqq 0$
while $\neg\,\text{Accept}(f, H, W) \wedge s \neq T$ **do**
  Pick $a \in \text{argmin}_W\, t_1, t_K$
  $t_a \coloneqq t_a + 1$
  Pull the arm a and receive the reward x_a
  $\hat{\mu}_a \coloneqq \left(1 - \frac{1}{t_a}\right)\hat{\mu}_a + \frac{1}{t_a}x_a$
  $H \coloneqq H \cap \{a \mid |x_a - \hat{\mu}_a| < c(t_a)\}$
  $W \coloneqq W \setminus \{b \mid \text{IndepTest}(f, H, b)\}$
  $s \coloneqq s + 1$
end while
return the single arm in $f(H)$.

The algorithm stops exploring an arm when it knows from a sensitivity-analysis subroutine IndepTest that, given the knowledge of the environment, the final decision will not change according to this arm. The termination of the algorithm is controlled by the predicate:

$$\text{Accept}(f, H, W) := \text{“}W = \varnothing\text{”} \implies |f(H)| = 1$$

2.4.5 Relative upper confidence bound (RUCB)

Zoghi et al. [2014a] consider the stochastic preference-based dueling bandits formulation. They extend the UCB algorithm and propose an algorithm called *Relative Upper Confidence Bound*(RUCB) provided that the preference matrix admits a Condorcet winner.

Algorithm 13 RUCB

Input: $A = \{1, \dots, K\}$, $\alpha > \frac{1}{2}, T \in \{1, 2, \dots\} \cup \infty$

$\mathcal{B} = \varnothing$ and $\boldsymbol{W} = [w_{ab}] \leftarrow 0_{K \times K}$ // w_{ab} is the number of times a beat b

for t=1, . . . , T **do**

  $\boldsymbol{U} := [u_{ab}] = \frac{\boldsymbol{W}}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}} + \sqrt{\frac{\alpha \log t}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}}}$ // all operations are element-wise

  $u_{aa} \leftarrow \frac{1}{2}$ for each $a \in A$ and $\mathcal{C} \leftarrow \{a \mid \forall b : u_{ab} \geq \frac{1}{2}\}$

  If $\mathcal{C} = \varnothing$, pick a randomly from A

  $\mathcal{B} \leftarrow \mathcal{B} \wedge \mathcal{C}$

  If $|\mathcal{C}| = 1$, then $\mathcal{B} \leftarrow \mathcal{C}$ and a to be the unique arm in $\mathcal{C}$.

  if $|\mathcal{C}| > 1$ **then**

    Sample a from $\mathcal{C}$ using the distribution:

$$p(a) = \begin{cases} 0.5 & \text{if } a \in \mathcal{B} \\ \frac{1}{2^{|\mathcal{B}|}|\mathcal{C} \setminus \mathcal{B}|} & \text{otherwise} \end{cases}$$

  end if

  $b \leftarrow \operatorname{argmax}_c u_{ca}$, with ties broken randomly. Moreover, if there is a tie, b is not allowed to be equal to a.

  Compare a and b and increment w_{ab} or w_{ba} depending on which arm wins.

end for

return An arm a that beats the most arms, i.e. a with the largest count $\#\left\{b \mid \frac{w_{ab}}{w_{ab}+w_{ba}} > \frac{1}{2}\right\}$

RUCB maintains upper confidence bound on the preference probabilities ff all possible pairs of arms. It then proceeds in two phases during which it chooses the two arms to select at the current time period. In the first round, an arm which beats all the other arms according to the optimistic preference estimates is selected as a

champion. If no such arm exists, a random arm is picked. In the second round, a normal classical bandit problem is set up using the preference estimates of all the arms against the champion selected in the first phase. The arm which has the highest preference estimate against the champion is selected as the competitor to the champion. Both of these arms or selected for a duel and based on which arm wins, the score sheet storing recording the results is updated which in turn affects the preference estimates for the next time period.

This algorithm achieves the upper bound of $\mathcal{O}(K \log T)$ on the expected cumulative Condorcet regret. Unlike the previous algorithms, RUCB is an anytime dueling bandits algorithm since it does not require the time horizon T as input.

2.4.6 Relative confidence sampling (RCS)

Like RUCB, the relative confidence sampling(RCS) algorithm proposed by Zoghi et al. [2014b] deals with the stochastic preference-based dueling bandit formulation. This algorithm is designed for the task of ranker evaluation on large-scale datasets. RCS too relies on the presence of a Condorcet winner in the preference matrix.

Algorithm 14 RCS

Input: $A = \{1, \dots, K\}$, $\alpha > \frac{1}{2}$
$\boldsymbol{W} = [w_{ab}] \leftarrow 0_{K \times K}$ // 2D array of wins: w_{ab} is the number of times a beat b
for $t = 1, 2, \dots$ **do**
    Phase I
    $\boldsymbol{\Theta}(t) \leftarrow \frac{\mathbf{1}_{K \times K}}{2}$
    for $a, b = 1, \dots, K$ with $a < b$ **do**
        $\boldsymbol{\Theta}_{ab}(t) \sim \text{Beta}(\boldsymbol{W}_{ab} + 1, \boldsymbol{W}_{ba} + 1)$
        $\boldsymbol{\Theta}_{ba}(t) = 1 - \boldsymbol{\Theta}_{ab}(t)$
    end for
    Pick c such that $\boldsymbol{\Theta}_{cb}(t) \geq 1/2$ for all b. If no such arm exists, pick the arm that has been chosen champion least frequently.
    Phase II
    $\boldsymbol{U} \leftarrow \frac{\boldsymbol{W}}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}} + \sqrt{\frac{\alpha \log t}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}}}$ // all operations are element-wise and division by zero is assumed to be zero.
    $\boldsymbol{U}_{aa} \leftarrow \frac{1}{2}$ for each $a \in A$
    $d \leftarrow \text{argmax}_b \boldsymbol{U}_{bc}(t)$
    Compare the arms c and d and increment either $\boldsymbol{W}_{cd}$ if c beat d or $\boldsymbol{W}_{dc}$ otherwise.
end for

This algorithm works in two phases. In the first phase, an arm deemed *champion* is elected by ways of a round-robin tournament based on previous arm comparisons.

In the second phase, this champion is compared against a worthy competitor. As time goes on, the best arm becomes increasingly likely to be the both the champion and the competitor, thereby causing the regret to fall steeply. A key characteristic of RCS is the use of sampling to conduct a round-robin tournament in the first phase. It maintains a Beta posterior distribution on $P_{a,b}$ for every pair of arms a, b. The samples from these posteriors are used to determine a champion arm c which beats all the other arms in this tournament. In the second phase, UCB is applied to the the classical bandit problem with mean rewards $\{P_{1,c}, \ldots, P_{K,c}\}$ to select the competitor d to duel with c.

RCS can be also used for the explore-then-exploit setting: when the horizon is reached, it picks any arm that has beat the greatest number of other arms at the final count.

2.4.7 Merge relative upper confidence bound (MERGERUCB)

Zoghi et al. [2015b] consider the problem of stochastic preference-based dueling bandit problem. This algorithm aims to avoid the quadratic dependence on the number of arms in the regret bound. For this purpose, they carry arm duels "locally" i.e. arms are placed in small batches that are processed separately and then merged together.

The proposed algorithm, (MERGERUCB), first groups the arms into small batches. Thereafter the algorithm proceeds in stages. During each stage, the arms within the same batch are compared against each other. The arms to be compared are chosen based on the upper confidence bounds on the preference probabilities. The current stage ends when the number of arms remaining becomes small and then pairs of batches are merged to form bigger batches. In the next stage, the same process repeats until a single arm remains.

MERGERUCB takes the initial size of each partition p and exploration parameter α. It also requires a parameter δ which is to be interpreted as the maximum probability of failure. With probability $1 - \delta$, it achieves the following cumulative Condorcet regret

$$\text{CRegret} \leq \frac{8\alpha p K \log T + C(\delta)}{\min_{a,b \,|\, \mathbb{P}_{a,b} \neq 0.5} \Delta_{ab}^2}$$

Algorithm 15 MERGERUCB

Input: $A = \{1, \dots, K\}$, the size of each partition $p \geq 4$, the maximum probability of failure δ, $\alpha > \frac{1}{2}$

$\boldsymbol{W} = [w_{ab}] \leftarrow 0_{K \times K}$ // 2D array of wins: w_{ab} is the number of times a beat b

$\mathcal{B}_1 = \left\{ \underbrace{\{1, \dots, p\}}_{B_1}, \dots, \underbrace{\{(b_1 - 1)p + 1, \dots, K\}}_{B_{b_1}} \right\}$ // a set of disjoint batches of rankers, with $b_1 = \left\lfloor \frac{K}{p} \right\rfloor$

$C(\delta) = \left\lceil \left(\frac{(4\alpha - 1)K^2}{(2\alpha - 1)\delta} \right)^{\frac{1}{2\alpha - 1}} \right\rceil$

$S = 1$ // the current stage of the algorithm

for t=1,2,... **do**

  $i = t \bmod b_S$

  $\boldsymbol{U} \leftarrow \frac{\boldsymbol{W}}{\boldsymbol{W} + \boldsymbol{W}^{\mathrm{T}}} + \sqrt{\frac{\alpha \log t + C(\delta)}{\boldsymbol{W} + \boldsymbol{W}^{\mathrm{T}}}}$ // all operations are element-wise

  For any $a \in B_i$ if $\boldsymbol{U}_{ab} < \frac{1}{2}$ for any $b \in B_i$, remove a from B_i.

  Select any $c \in B_i$ randomly.

  Set $d \leftarrow \operatorname{argmax}_{\{a | a \in B_i \setminus \{c\}\}} \boldsymbol{U}_{ac}$

  Compare the arms c and d and increment either $\boldsymbol{W}_{cd}$ if c beat d or $\boldsymbol{W}_{dc}$ otherwise.

  if $\sum_i |B_i| \leq \frac{K}{2^T}$ **then**

    Combine pairs of batches of arms so that each new batch has between $p/2$ and $3p/2$ arms in it, pairing the smallest batches with the largest ones, making sure that each batch contains at least two arms. Update the sets B_i, putting them all in the set $\mathcal{B}_S$, and define $b_S := |\mathcal{B}_S|$

    S = S + 1

  end if

end for

where $C(\delta) = \left\lceil \left(\frac{(4\alpha - 1)K^2}{(2\alpha - 1)\delta} \right)^{\frac{1}{2\alpha - 1}} \right\rceil$. By setting $\delta = 1/T$, an upper confidence bound on the expected cumulative regret can be achieved too. If $\alpha \geq 1$, the expected cumulative regret of MERGERUCB is upper bounded by $O(K \log T)$

2.4.8 Copeland confidence bound (CCB)

Zoghi et al. [2015a] consider the stochastic preference-based dueling bandit problem in which a Condorcet winner might not exist. Instead they propose an algorithm called *Copeland confidence bound* (CCB) which aims to minimize the cumulative regret with respect to the Copeland winner, which unlike the Condorcet winner, is guaranteed to exist.

Algorithm 16 CCB

Input: $A = \{1, \ldots, K\}$, $\alpha > \frac{1}{2}$

1: $\boldsymbol{W} = [w_{ab}] \leftarrow 0_{K \times K}$ // 2D array of wins: w_{ab} is the number of times a beat b

2: $\mathcal{B}_1 = \{1, \ldots, K\}$ //potential best arms

3: $\mathcal{B}_1^a = \varnothing$ for each $a = 1, \ldots, K$ //potential to beat the arm i

4: $\overline{L}_C = K$ // estimated max losses of a Copeland winner

5: **for** $t = 1, 2, \ldots,$ **do**

6: $\boldsymbol{U} \coloneqq [u_{ab}] = \frac{\boldsymbol{W}}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}} + \sqrt{\frac{\alpha \log t}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}}}$ and $\boldsymbol{L} \coloneqq [l_{ab}] = \frac{\boldsymbol{W}}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}} - \sqrt{\frac{\alpha \log t}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}}}$, with $u_{ab} = l_{ab} = \frac{1}{2}, \forall a$

7: $\overline{\mathrm{Cpld}}(a) = \#\{b \mid u_{ab} \geq \frac{1}{2}, b \neq a\}$ and $\underline{\mathrm{Cpld}}(a) = \#\{b \mid l_{ab} \geq \frac{1}{2}, b \neq a\}$

8: $\mathcal{C}_t = \{a \mid \overline{\mathrm{Cpld}}(a) = \max_b \overline{\mathrm{Cpld}}(b)\}$

9: Set $\mathcal{B}_t \leftarrow \mathcal{B}_{t-1}$ and $\mathcal{B}_t^a = \mathcal{B}_{t-1}^a$ and update as follows:

A. Reset disproven hypotheses: If for any a and $b \in \mathcal{B}_t^a$ we have $l_{ab} > 0.5$, reset $\mathcal{B}_t$, $\overline{L}_C$ and B_t^c for all the c.

B. Remove non-Copeland winners: For each $a \in B_t$, if $\overline{\mathrm{Cpld}}(a) < \underline{\mathrm{Cpld}}(b)$ holds for any b, set $\mathcal{B}_t \leftarrow \mathcal{B}_t \setminus \{a\}$, and if $|B_t^a| \neq \overline{L}_C + 1$ then set $\mathcal{B}_t^a \leftarrow \{c \mid u_{ac} < 0.5\}$. However if $\mathcal{B}_t = \varnothing$, reset $\mathcal{B}_t, \overline{L}_C$ and B_t^c for all c.

C. Add Copeland winners: For any $a \in \mathcal{C}_t$ with $\overline{\mathrm{Cpld}}(b) = \underline{\mathrm{Cpld}}(a)$, set $\mathcal{B}_t \leftarrow \mathcal{B}_t \cup \{a\}$, $\mathcal{B}_t^a \leftarrow \varnothing$ and $\overline{L}_C \leftarrow K - 1 - \overline{\mathrm{Cpld}}(a)$. For each $b \neq a$, if we have $|B_t^b| < \overline{L}_C + 1$, set $\mathcal{B}_t^a \leftarrow \varnothing$, and if $|B_t^b| < \overline{L}_C + 1$, randomly choose $\overline{L}_C + 1$ elements of B_t^b and remove the rest.

10: With probability $1/4$, sample (c, d) uniformly from the set $\{(a, b) \mid b \in B_t^a$ and $0.5 \in [l_{ab}, u_{ab}]\}$ (if it is non-empty) and skip to Line 14.

11: If $\mathbb{B}_t \cap \mathcal{C}_t \neq \varnothing$, then with probability $2/3$, set $\mathcal{C}_t \leftarrow \mathcal{B}_t \cap \mathcal{C}_t$.

12: Sample c from C_t uniformly at random.

13: With probability $1/2$, choose the set $\mathcal{B}^a$ to be either $\mathcal{B}_t^a$ or $\{1, \ldots, K\}$ and then set $d \leftarrow \mathrm{argmax}_{b \in B^a \mid l_{bc} \leq 0.5} u_{bc}$. If there is a tie, d is not allowed to be equal to c.

14: Compare the arms c and d and increment w_{cd} or w_{dc} depending on which arm wins.

15: **end for**

2.4.9 Contextual dueling bandits

Dudík et al. [2015] consider the problem of stochastic preference-based dueling bandit problem. The two salient features of this article are the introduction of the *Von Neumann winner* and incorporating context in the dueling bandit problem. Instead of containing the preference probabilities, they modify the preference matrix to contain the expectations of outcomes[5] of duels between all the pair of actions.

$$\psi(a, b) = \begin{cases} +1 & a \text{ wins the duel} \\ -1 & b \text{ wins the duel} \end{cases}$$

[5]Here we overload the definition of outcome which is slightly different from the previous definition given in equation 2.4 where it can take the value 0 too.

The authors assume no ties. Of course, $\psi(a, b)$ and preference matrix element $P_{a,b}$ are related as follows:

$$P_{a,b} = \frac{\psi(a, b) + 1}{2}$$

With this construction of the preference matrix P, a Von Neumann winner is defined as a probability vector $\boldsymbol{w}$ in the simplex vectors in $[0, 1]^K$ whose entries sum to 1 such that

$$\sum_{a=1}^{K} w(a) P_{a,b} \geq 0 \quad \text{for all actions } b$$

Like a Condorcet winner, a Van Neumann winner has at least 50% chance of wining against any other policy. However, unlike a Condorcet winner, a Von Neumann winner is guaranteed to exist provided that the preference matrix P is skew symmetric which implies that a duel (b, a) is equivalent to the negation of a duel (a, b), as is natural.

Furthermore in this setting, the learner is allowed to observe a context selected by the environment. The authors provide two algorithms to compute an approximation of a Von Neumann winner for the explore-then-exploit version of this problem. The first algorithm, SPARRINGFPL, is based on the Follow-the-Perturbed-Leader (FPL) algorithm of Kalai and Vempala [2005]. The second algorithm, PROJECTGD, is based on online projected gradient descent methods of Zinkevich [2003b].

2.4.10 Double Thompson sampling for dueling bandits

Wu and Liu [2016] consider the problem stochastic preference-based dueling bandit problem and they propose an algorithm called Double Thompson Sampling (D-TS). D-TS maintains a posterior distribution for the preference matrix, and chooses the pair of arms at each time period according to the two set of samples drawn independently from the posterior distribution. For Copeland dueling bandits, D-TS achieves the cumulative regret bound of $O(K^2 \log T)$.

Algorithm 17 D-TS for Copeland dueling bandits

Input: $A = \{1, \ldots, K\}$, $\alpha > \frac{1}{2}$
$\boldsymbol{W} = [w_{ab}] \leftarrow 0_{K \times K}$ // 2D array of wins: w_{ab} is the number of times a beat b
for $t = 1, \ldots, T$ **do**
// Phase I: Choose the first arm $a^{(1)}$
  $\boldsymbol{U} \coloneqq [u_{ab}] = \frac{\boldsymbol{W}}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}} + \sqrt{\frac{\alpha \log t}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}}}$ and $\boldsymbol{L} \coloneqq [l_{ab}] = \frac{\boldsymbol{W}}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}} - \sqrt{\frac{\alpha \log t}{\boldsymbol{W}+\boldsymbol{W}^{\mathrm{T}}}}$, with $u_{ab} = l_{ab} = \frac{1}{2}, \forall a$ *// Division by zero is assumed to be zero.*
  $\mathrm{UB}_a \leftarrow \frac{1}{K-1} \sum_{b \neq a} \mathbb{1}(u_{ab} > 1/2)$ *// Upper bound on the normalized Copeland score.*
  $C \leftarrow \{a : \mathrm{UB}_a = \max_b \mathrm{UB}_b\}$
  for $a, b = 1, \ldots, K$ with $a < b$ **do**
    Sample $\theta^{(1)}_{ab} \sim \mathrm{Beta}(w_{ab} + 1, w_{ba} + 1)$
    $\theta^{(1)}_{ba} \leftarrow 1 - \theta^{(1)}_{ab}$
  end for
  $a^{(1)} \leftarrow \mathrm{argmax}_{a \in C} \sum_{b \neq a} \mathbb{1}_{(\theta^{(1)}_{ab} > 1/2)}$ *// Choosing from C to eliminate likely non-winner arms with ties broken randomly.*
// Phase II: Choose the second arm.
  Sample $\theta^{(2)}_{ba^{(1)}} \sim \mathrm{Beta}(w_{ba^{(1)}}+1, w_{a^{(1)}b}+1)$ for all $b \neq a^{(1)}$, and let $\theta^{(2)}_{a^{(1)}a^{(1)}} = 1/2$
  $a^{(2)} \leftarrow \mathrm{argmax}_{b: l_{ba^{(1)}} \leq 1/2} \theta^{(2)}_{ba^{(1)}}$ *// Choosing only from uncertain pairs*
  Compare the pair $(a^{(1)}, a^{(2)})$ and increment $w_{a^{(1)}a^{(2)}}$ if $a^{(1)}$ wins or otherwise increment $w_{a^{(2)},a^{(1)}}$
end for

2.4.11 SPARRING

Ailon et al. [2014] reduce the stochastic utility-based dueling bandits problem to the conventional MAB problem. They use the notion of bandit regret. They propose an algorithm called SPARRING which uses two separate classical bandit algorithms (CBA), each for one arm, to choose the pair of arms to be played at every time period.

Algorithm 18 SPARRING

Input: $A = \{1, \ldots, K\}$
LCBA, RCBA $\leftarrow$ Two classical bandit algorithms over A
LCBA.`init`(), RCBA.`init`(), $t \leftarrow 1$
while true **do**
  $a_t \leftarrow$ LCBA.`decide`(); $b_t \leftarrow$ RCBA.`decide`()
  Play (a_t, b_t) and observe $y_t \in \{0, 1\}$
  LCBA.`set_feedback`($\mathbb{1}_{(y_t=0)}$); RCBA.`set_feedback`($\mathbb{1}_{(y_t=1)}$)
  $t \leftarrow t + 1$
end while

Each CBA has three subroutines: `init`(), `decide`() and `feedback`(). The `init`() subroutine simply clears its state. The `decide`() subroutine returns the next arm to play and the `set_feedback`() subroutine provides the feedback to the algorithm.

SPARRING algorithm, although originally designed for stochastic settings, can work for adversarial setting as well with an algorithm EXP3 used as CBA. It preserves the $\mathcal{O}(\sqrt{KT \ln K})$ upper bound of EXP3.

In Table 2.1, we provide a comparative summary of all the algorithms studied in this chapter.

TABLE 2.1: Summary of dueling bandit algorithms

Algorithm	Rewards	Formulation	Settings
DBGD	Stochastic	Utility-based	Exploration-exploitation
IF	Stochastic	Preference-based	Exploration-exploitation
BTM	Stochastic	Preference-based	PAC & Exploration-exploitation
SAVAGE	Stochastic	Preference-based	Exploration-exploitation
RUCB	Stochastic	Preference-based	Exploration-exploitation
RCS	Stochastic	Preference-based	Exploration-exploitation
MERGERUCB	Stochastic	Preference-based	Exploration-exploitation
DTS	Stochastic	Preference-based	Exploration-exploitation
CCB	Stochastic	Preference-based	Exploration-exploitation
SPARRING	Stochastic	Utility-based	Exploration-exploitation

With this, we conclude the overview of the related work on dueling bandits. In the next chapter, we shall see what is the best possible performance any dueling bandit algorithm can achieve in the exploration-exploitation setting.

Chapter 3

The Lower Bound

In this short chapter, we shall prove the lower bound on the cumulative regret of any dueling bandit algorithm in the exploration-exploitation setting.

To provide a lower bound on the regret of any dueling bandits algorithm, we use a reduction to the classical MAB problem suggested by Ailon et al. [2014]. Algorithm 19 gives an explicit formulation of this reduction by using a generic dueling bandits algorithm (DBA) as a black-box having the following public sub-routines: `init`(), `decide`() and `feedback`(). The subroutine `init`() is used to initialize the algorithm, `decide`() returns the pair of arms to be pulled at any given time instant and `set_feedback`() provides the relative feedback to the algorithm. The classical bandit environment (CBE) provides `get_reward`() which returns the reward of the input arm.

Algorithm 19 Reduction to classical MAB

Input: $A = \{1, \ldots, K\}$
DBA.`init`()
Set $t = 1$
repeat
  $(a_t, b_{t+1}) \leftarrow$ DBA.`decide`()
  $x_{a_t} \leftarrow$ CBE.`get_reward`()
  $x_{b_{t+1}} \leftarrow$ CBE.`get_reward`()
  DBA.`set_feedback`$((\mathtt{a_t}, \mathtt{b_{t+1}}), (\mathtt{x_{a_t}} - \mathtt{x_{b_{t+1}}}))$
  $t = t + 2$
until $t \geq T$

Let us consider that the reward at time t in the dueling bandits setting is the mean of the individual rewards of the chosen arms at that time. This was defined earlier in equation (2.1). Therefore the reward obtained by algorithm 19 on selection of the pair (a_t, b_{t+1}) is $\frac{x_{a_t} + x_{b_t+1}}{2}$. On the other hand, in the classical bandit setting, the reward

obtained on selection of arm a_t followed by arm b_{t+1} is $x_{a_t} + x_{b_t+1}$. So clearly the expected classical-bandit reward of Algorithm 19 will be twice to expected reward of the black-box dueling bandit algorithm it uses. Consequently, the expected regret of DBA is of the order of the expected regret in classical bandit setting.

It is important to note that this reduction only works for stochastic settings where the expected reward of each arm remains the same across time instants because the rewards are drawn from stationary distributions. According to Theorem 5.1 given by Auer et al. [2002b, section 5], for $K \geq 2$, the expected regret in the classical bandit setting is $\Omega(\sqrt{KT})$ (assuming T is large enough i.e. $T \geq \sqrt{KT}$). Since this result is obtained with a stationary stochastic distribution, by extension, the expected regret for any dueling bandits setting cannot be less than $\Omega(\sqrt{KT})$. Therefore we can have have a lower bound on the expected regret for dueling bandits as follows:

Theorem 3.1. *For any number of actions $K \geq 2$ and large enough time horizon T (i.e. $T \geq \sqrt{KT}$), there exists a distribution over assignments of rewards such that the expected cumulative regret of any utility-based dueling bandits algorithm cannot be less than $\Omega(\sqrt{KT})$.*

After having the lower bound, the next step for us is to devise an algorithm for the dueling bandit problem in the exploration-exploitation setting. We provide the same in the next chapter.

Chapter 4

The Algorithm and its Analysis

In Section 4.1, we introduce the *Relative Exponential-weight Algorithm for Exploration and Exploitation*(REX3). The implementation of the algorithm on a simple toy problem is illustrated in Section 4.2. Lastly, we prove a finite-horizon upper bound on the cumulative regret of REX3 in Section 4.3.

4.1 Relative Exponential-weight Algorithm for Exploration and Exploitation (REX3)

We propose an algorithm for dueling bandits in the exploration-exploitation setting. The pseudo-code for the algorithm we propose, called Relative Exponential-weight Algorithm for Exploration and Exploitation (REX3) is given in Algorithm 20. This algorithm is designed to apply for the adversarial utility-based dueling bandits problem.

It is similar to the original EXP3 from step 1 to step 6 where it computes a distribution $\mathbf{p}(t) = (p_1(t), \ldots, p_K(t))$ which is a mixture of a normalized weighing of the arms $w_i / \sum_i w_i$ and a uniform distribution $1/K$. As in EXP3, this uniform probability is introduced to ensure a minimum exploration of all arms.

At step 7, the algorithm draws two arms a and b independently according to $\mathbf{p}(t)$. At step 8, the algorithm gets $\psi(x_a - x_b)$ as relative feedback . Note that, since arms are drawn with replacement, we may have $a = b$, in which case the algorithm will get no information. This event is indeed expected to become frequent when the $\mathbf{p}(t)$ distribution becomes peaked around the best arms. This necessity for a regret-minimizing dueling bandits algorithm to renounce getting information when confident about its

Algorithm 20 REX3

Input: $A = \{1, \dots, K\}$
Parameters: $\gamma \in (0, 1]$
1: **Initialization:** $w_i(1) = 1$ for $i = 1, \dots, K$.
2: **for** $t = 1, 2, \dots$ **do**
3: **for** $i = 1, \dots, K$ **do**
4: Set $p_i(t) \leftarrow (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$
5: **end for**
6: Pull two arms a and b chosen independently according to the distribution $(p_1(t), \dots, p_K(t))$.
7: Receive relative feedback $\psi(x_a - x_b) \in [-1, +1]$
8: **if** $a \neq b$ **then**
9: Set $w_a(t+1) \leftarrow w_a(t) \cdot e^{\frac{\gamma}{K}\frac{\psi(x_a - x_b)}{2p_a}}$
10: Set $w_b(t+1) \leftarrow w_b(t) \cdot e^{-\frac{\gamma}{K}\frac{\psi(x_a - x_b)}{2p_b}}$
11: **end if**
12: *Update γ (for anytime version)*
13: **end for**

decision is a structural bias toward exploitation that is not encountered in classical bandits.

Step 8 is the big difference from EXP3; because we only have access to the relative $\psi(x_a - x_b)$ value, we have no mean to estimate the individual rewards x_a or x_b. There is however a solution to circumvent this problem: the best arm in expectation at time t is not only the one which maximizes the absolute reward. It is indeed the one which maximizes the regret of any fixed strategy $\pi(t)$ against it:

$$\operatorname*{argmax}_i x_i(t) = \operatorname*{argmax}_i \left(x_i(t) - \mathbb{E}_{a \sim \pi(t)} x_a \right) .$$

This reference strategy could be a single-arm or uniform strategy but playing a sub-optimal strategy to get a reference has a cost in terms of regret. One of our contributions is to show that the algorithm may use its own strategy as a reference.

At step 9, the condition $a \neq b$ is only a slight improvement for preference-based dueling bandits where the outcome of a duel of an arm against itself is randomized as in Eq. (2.5).

At steps 10 and 11, the weights of the played arms are updated. This update process is the core of our algorithm, it will be detailed in Section 4.3.

Step 13 is only required for the anytime version of the algorithm. It will be explained in Section 5.2.

4.2 Illustration of REX3 on a toy problem

To understand the working of this algorithm, let us see how it performs on a 4-armed dueling bandit problem with the first arm being optimal. Let us assume $\gamma = 0.4$ and ψ to be an identity function. Initially, the weights assigned for all the arms are all equal to 1, as shown in figure 4.1. Therefore at time instant $t = 1$, the probability of any of the fours arms being selected is the same i.e.

$$p_1(1) = p_2(1) = p_3(1) = p_4(1) = 0.25$$

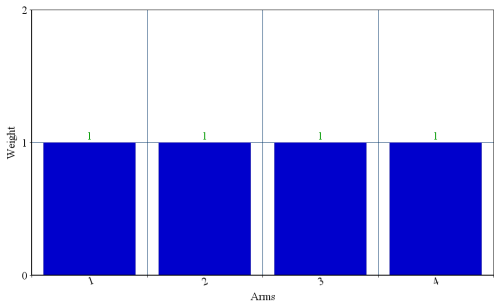


FIGURE 4.1: REX3: weights at $t = 1$

Let the arms being picked by the algorithm at time instant $t = 1$ be 1 and 2, hence

$$a = 1 \text{ and } b = 1$$

Depending upon the rewards of these arms, either the first arm or the second one wins this duel. Let the reward set by the adversary for arm 1 be greater than that of 2 i.e. $x_a > x_b$. Since we are dealing with binary rewards, that translates to $x_a - x_b = 1$. Let us now see how REX3 computes the weights for all the arms at $t = 2$. The weights of the arms not selected by the algorithm during the previous time period remain the same, hence

$$w_3(2) = w_3(1) = 1$$

$$w_4(2) = w_4(1) = 1$$

The weights of the arms selected at $t = 1$, however, are computed using the update rule as follows:

$$w_1(2) \leftarrow w_1(1) \cdot e^{\frac{0.4}{4}\frac{1}{0.5}} \approx 1.22$$
$$w_2(2) \leftarrow w_2(1) \cdot e^{\frac{0.4}{4}\frac{-1}{0.5}} \approx 0.82$$

These weights are depicted in a bar chart in figure 4.2

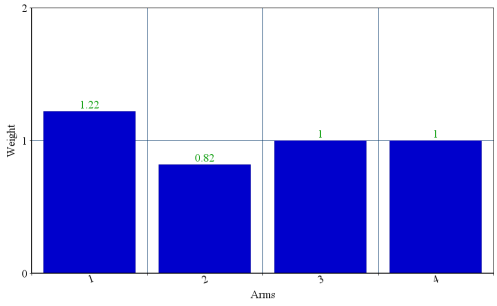


FIGURE 4.2: REX3: weights at $t = 2$

Using these weights, the logarithm computes the distribution $\boldsymbol{p}_t$ as follows:

$$p_1(2) \leftarrow (1 - 0.4)\frac{1.22}{4.04} + \frac{0.4}{4} \approx 0.28$$
$$p_2(2) \leftarrow (1 - 0.4)\frac{0.82}{4.04} + \frac{0.4}{4} \approx 0.22$$
$$p_3(2) \leftarrow (1 - 0.4)\frac{1}{4.04} + \frac{0.4}{4} \approx 0.25$$
$$p_4(2) \leftarrow (1 - 0.4)\frac{1}{4.04} + \frac{0.4}{4} \approx 0.25$$

Hence arm 1 winning the duel results in its probability of being selected during the next time period being increased while on the other hand, the probability of arm 2 being selected during the next time period is reduced because it lost the duel. The algorithms then proceeds to draw a and b according to $\boldsymbol{p}(2)$. Let $a = 1$ and $b = 3$. Assume that 1 wins this duel too because the adversary has set the reward for the first arm higher than that of third arm during $t = 2$. Therefore $x_a - x_b = 1$. The algorithm then computes the weights for each of the fours arms. The weights of the

arms not selected at $t = 2$ are not affected, hence

$$w_2(3) = w_2(2) = 0.82$$
$$w_4(3) = w_4(2) = 1$$

The weights of 1 and 3 are computed as follows:

$$w_1(3) \leftarrow w_1(2) \cdot e^{\frac{0.4}{4}\frac{1}{0.56}} \approx 1.45$$
$$w_3(3) \leftarrow w_3(2) \cdot e^{\frac{0.4}{4}\frac{-1}{0.5}} \approx 0.82$$

These weights are shown in the form of a bar chart in figure 4.3

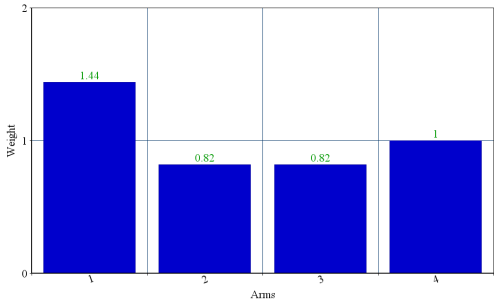


FIGURE 4.3: REX3: weights at $t = 3$

So we see how the algorithm increases the weight of an arm which wins the duel and correspondingly also decreases the weight of an arm which loses the duel. The better arms are likely to win more duels than the worse arms and hence the weights corresponding to the former are progressively increased by the algorithm after each duel. This results in higher weights for the better arms. Higher the weight, better is the probability of the corresponding arm being selected by the algorithm and hence the algorithm selects better arms with higher and higher probability as they establish their performance superiority over the worse arms.

4.3 Upper bound on the regret of REX3

In this section, we provide a finite-horizon upper bound on the expected regret of REX3. For the analysis, we focus on the simple case where ψ is the identity. It

provides a ternary *win/tie/loss* feedback if we assume binary rewards as follows:

$$\psi(x_a, x_b) = \begin{cases} 1 & \text{if } x_a = 1 \wedge x_b = 0 \\ 0 & \text{if } x_a = x_b = 0 \vee x_a = x_b = 1 \\ -1 & \text{if } x_a = 0 \wedge x_b = 1 \end{cases}$$

The main difference between EXP3 and our algorithm is at steps 10 and 11 of Algorithm 20, where we update the weights according to the duel outcome: the winning arm is gratified while the loser is penalized. This 'punitive' approach of exponential weighing departs from EXP3 and other weighing algorithms which gratify the most rewarding arms while kindly ignoring the non-rewarding ones (Freund and Schapire [1999b], Cesa-Bianchi and Lugosi [2006]).

The steps 10-11 on Algorithm 20 are equivalent to operating for each arm i an update of the form:

$$w_i(t+1) = w_i(t) \cdot e^{\frac{\gamma}{K}\hat{c}_i(t)}$$

where

$$\hat{c}_i(t) = \mathbb{1}_{(i=a)} \frac{\psi(x_a - x_b)}{2p_a} + \mathbb{1}_{(i=b)} \frac{\psi(x_b - x_a)}{2p_b} \tag{4.1}$$

One big difference with EXP3 is that $\hat{c}_i(t)$ is not an estimator of the reward $x_i(t)$. We instead have:

Lemma 4.1.

$$\mathbb{E}\left[\hat{c}_i(t) | (a_1, b_1), .., (a_{t-1}, b_{t-1})\right] = \mathbb{E}_{a \sim p(t)} \psi(x_i(t) - x_a(t))$$

Proof.

$$\begin{aligned} \hat{c}_i(t) &= \mathbb{1}_{(i=a_t)} \frac{\psi(x_{a_t} - x_{b_t})}{2p_{a_t}(t)} + \mathbb{1}_{(i=b_t)} \frac{\psi(x_{b_t} - x_{a_t})}{2p_{b_t}(t)} \\ \mathbb{E}_{(a,b)\sim p(t)} \hat{c}_i(t) &= \sum_{j=1}^{K} \sum_{k=1}^{K} p_j(t) p_k(t) \left(\mathbb{1}_{(i=j)} \frac{\psi(x_j - x_k)}{2p_j} + \mathbb{1}_{(i=k)} \frac{\psi(x_k - x_j)}{2p_k} \right) \\ &= \sum_{j=1}^{K} \sum_{k=1}^{K} p_j p_k \mathbb{1}_{(i=j)} \frac{\psi(x_j - x_k)}{2p_j} + \sum_{j=1}^{K} \sum_{k=1}^{K} p_j p_k \mathbb{1}_{(i=k)} \frac{\psi(x_k - x_j)}{2p_k} \end{aligned}$$

$$= \frac{1}{2}\sum_{k=1}^{K} p_k \psi(x_i - x_k) + \frac{1}{2}\sum_{j=1}^{K} p_j \psi(x_i - x_j)$$

$$= \mathbb{E}_{a\sim p}\psi(x_i - x_a)$$

If ψ is identity, it simplifies into:

$$\mathbb{E}_{(a,b)\sim p(t)}\hat{c}_i(t) = \quad x_i - \mathbb{E}_{a\sim p(t)}x_a$$

□

If ψ is the identity then $\mathbb{E}\hat{c}_i(t) = x_i(t) - \mathbb{E}_{a\sim p(t)}x_a(t)$ in which case we estimate the expected instantaneous regret of the algorithm against arm i. If we rather take $\psi(x) = \mathbb{1}_{(x>0)}$, then $\mathbb{E}\hat{c}_i(t) = \mathbb{P}_{a\sim p(t)}\left(x_i(t) > x_a(t)\right)$, i.e. the probability for the algorithm to select an arm defeated by i.

Let $\mathbb{G}_{max} = \max_i \sum_{t=1}^{T} x_i(t)$ be the best single-arm gain, and let $\mathbb{G}_{alg} = \frac{1}{2}\sum_{t=1}^{T} x_a(t) + x_b(t)$ be the gain of the algorithm. Let $\mathbb{E}\mathbb{G}_{unif} = \frac{1}{K}\sum_{t=1}^{T}\sum_{i=1}^{K} x_i(t)$ be the average value of the game (i.e. the expected gain of the uniform sampling strategy).

Theorem 4.1. *If the transfer function ψ is the identity and $\gamma \in (0, \frac{1}{2})$, then,*

$$\mathbb{G}_{max} - \mathbb{E}(\mathbb{G}_{alg}) \leq \frac{K}{\gamma}\ln(K) + \gamma\tau$$

where $\tau = e \cdot \mathbb{E}\mathbb{G}_{alg} - (4-e) \cdot \mathbb{E}\mathbb{G}_{unif}$.

Proof sketch: The general structure of the proof is similar to the one of [Auer et al., 2002b, section 3], but, as explained before, the $\hat{c}_i(t)$ estimator we use differs from the one of EXP3 because it gives an instantaneous regret estimate instead of an absolute reward estimate. As such, it may reach negative values and the $w_i(t)$ weights may decrease with time. We only give here a sketch of proof, stressing on the differences from Auer et al. [2002b]. The complete step-by-step proof is deferred til Appendix A.

Let $W_t = w_1(t) + w_2(t) + \cdots + w_K(t)$. As in EXP3 proof we consider:

$$\frac{W_{t+1}}{W_t} = \sum_{i=1}^{K} \frac{p_i(t) - \gamma/K}{1-\gamma} e^{(\gamma/K)\hat{c}_i(t)}$$

The inequality $e^x \leq 1 + x + (e-2)x^2$ is tight for $x \in [0,1]$ but it remains valid for negative values, hence:

$$\frac{W_{t+1}}{W_t} \leq 1 - \frac{\gamma^2/K}{1-\gamma}\left(\underbrace{\frac{1}{K}\sum_{i=1}^{K}\hat{c}_i(t)}_{=-M_1}\right) + \frac{(e-2)\gamma^2/K}{1-\gamma}\left(\underbrace{\frac{1}{K}\sum_{i=1}^{K}p_i(t)\hat{c}_i(t)^2}_{=M_2}\right)$$

As in EXP3 we take the logarithm and sum over t. We get for any j:

$$\sum_{t=1}^{T}\frac{\gamma}{K}\hat{c}_j(t) - \ln(K) \leq \frac{\gamma^2/K}{1-\gamma}M_1 + \frac{(e-2)\gamma^2/K}{1-\gamma}M_2$$

By taking the expectation over the algorithm's randomization, we obtain for any j:

$$\sum_{t=1}^{T}\frac{\gamma}{K}\underbrace{\mathbb{E}_{\sim p}\hat{c}_j(t)}_{(4.3)} - \ln(K) \leq \frac{\gamma^2/K}{1-\gamma}\sum_{i=t}^{T}\underbrace{\mathbb{E}_{\sim p}M_1}_{(4.4)} + \frac{(e-2)\gamma^2/K}{1-\gamma}\sum_{i=t}^{T}\underbrace{\mathbb{E}_{\sim p}M_2}_{(4.5)} \tag{4.2}$$

From Lemma 4.1 we directly get the expected regret against j on the left side of the inequality:

$$\mathbb{E}_{\sim p}\hat{c}_j(t) = x_j - \mathbb{E}_{\sim p}(x_a) \tag{4.3}$$

By averaging (4.3) over the arms, we obtain:

$$\mathbb{E}_{\sim p(t)}M_1 = -\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{\sim p}\hat{c}_i(t) = \mathbb{E}(x_a) - \frac{1}{K}\sum_{i=1}^{K}x_i \tag{4.4}$$

The following result too is detailed in Appendix A:

$$\mathbb{E}_{\sim p(t)}M_2 \leq \frac{1}{2}\mathbb{E}(x_a) + \frac{1}{2K}\sum_{i=1}^{K}x_i \tag{4.5}$$

From Lemma 4.1, Equation (4.4) and Inequality (4.5), and by definition of $\mathbb{G}_{max}$, $\mathbb{EG}_{alg}$, and $\mathbb{EG}_{unif}$, the Inequality (4.2) rewrites into:

$$\mathbb{G}_{max} - \mathbb{EG}_{alg} - \frac{K\ln K}{\gamma} \;\leq\; \frac{\gamma}{1-\gamma}\left(\mathbb{EG}_{alg} - \mathbb{EG}_{unif}\right) + \frac{(e-2)\gamma}{2(1-\gamma)}\left(\mathbb{EG}_{alg} + \mathbb{EG}_{unif}\right)$$

Assuming $\gamma \leq \frac{1}{2}$, we finally obtain:

$$\mathbb{G}_{max} - \mathbb{E}\mathbb{G}_{alg} \leq \frac{K \ln K}{\gamma} + \gamma \left(e\mathbb{E}\mathbb{G}_{alg} - (4-e)\, \mathbb{E}\mathbb{G}_{unif}\right)$$

□

Provided that $\mathbb{E}\mathbb{G}_{alg} \leq \mathbb{G}_{max}$ and $\mathbb{E}\mathbb{G}_{unif} \geq \mathbb{G}_{min}$, where $\mathbb{G}_{min} = \min_i \sum_{t=1}^{T} x_i(t)$ is the gain of the worst single-arm strategy, we can simplify the bound into:

Corollary 4.1. $\mathbb{G}_{max} - \mathbb{E}\mathbb{G}_{alg} \leq \frac{K \ln K}{\gamma} + \gamma \left(e\mathbb{G}_{max} - (4-e)\, \mathbb{G}_{min}\right)$

As in [Auer et al., 2002b, section 3], since $\frac{K}{\gamma} \ln(K) + \gamma\tau$ is convex, we can obtain the optimal γ on $(0, \frac{1}{2})$:

$$\gamma* = \min \left\{ \frac{1}{2}, \sqrt{\frac{K \ln(K)}{\tau}} \right\} \tag{4.6}$$

Substituting γ in Corollary 4.1 with its optimal value from eq. (4.6) we obtain:

$$\mathbb{G}_{max} - \mathbb{E}(\mathbb{G}_{alg}) \leq 2\sqrt{K \ln(K) \left[e\mathbb{G}_{max} - (4-e)\, \mathbb{G}_{min}\right]}$$

Hence,

Corollary 4.2. *When* $\gamma = \min \left\{ \frac{1}{2}, \sqrt{\frac{K \ln(K)}{\tau}} \right\}$, *the expected regret of* REX3 *(Algorithm 20) is* $\mathcal{O}\left(\sqrt{K \ln(K) T}\right)$.

The upper bound of REX3 for adversarial utility-based dueling bandits is hence the same as the one of EXP3 for classical adversarial MABs. This is remarkable as the relative feedback in the dueling bandits can be considered as a more restrictive since relative feedback For a high-number of arms or a short term horizon, this bound is competitive against the $\mathcal{O}\left(K \ln(T)\right)$ or $\mathcal{O}\left(K^2 \ln(T)\right)$ existing bounds for stochastic dueling bandits.

This corollary brings us to the end of this chapter. In the next chapter, we shall see how the empirical performance of REX3 on real datasets compares against the state of the art algorithms.

Chapter 5

Empirical Evaluation

In the first part of this chapter, given in Section 5.1, we provide the experimental results to verify Corollary 4.1. In the second part of this chapter, given in Section 5.2, we compare the performance of REX3 to the state of the art algorithms which were introduced in Section 2.4.

To evaluate REX3 and other dueling bandits algorithms, we have applied them to the online comparison of rankers for search engines by *interleaved filtering* Radlinski and Joachims [2007]. A search-engine ranker is a function that orders a collection of documents according to their relevancy to a given user search query. By interleaving the output of two rankers and tracking on which ranker's output the user did click, we are able to get an unbiased feedback about the relative quality of these two rankers. Given K rankers, the problem of finding the best ranker is indeed a K-armed dueling bandits.

In order to obtain reproducible and comparable results, we adopted the stochastic preference-based experiment setup already employed by Yue and Joachims [2011], Zoghi et al. [2014a,c, 2015c] with both the cumulative Condorcet regret as defined by Yue et al. [2012], Urvoy et al. [2013a] and the *accuracy* i.e. the best arm selection-rate over the runs.

This experimental setup uses real search engines' logs to build empirical preference matrices. We used several preference matrices issued from namely: ARXIV dataset (Yue and Joachims [2011]), LETOR NP2004 dataset (Liu et al. [2007]), and MSLR30K dataset(MSLR30K [2012]). The last dataset distinguishes three kinds of queries: informational, navigational and perfect-hit navigational. These matrices are courtesy the authors of Zoghi et al. [2014c]. The preference matrices we used

TABLE 5.1: The preference matrices used for experiments

Dataset	K	Condorcet	=Borda?
ARXIV 2011	6	yes	yes
LETOR NP2004	16	yes	yes
LETOR NP2004	32	yes	yes
LETOR NP2004	64	yes	yes
MSLR INF.	136	**no**	-
MSLR NAV.	136	yes	yes
MSLR PERF.	136	yes	yes
SAVAGE (artificial)	6	yes	yes
SAVAGE (artificial)	30	yes	yes
BVS (artificial)	20	yes	**no**

and their properties are summarized in Table 5.1.

5.1 Empirical validation of Corollary 4.1

We have used LETOR NP2004 and MSLR30K datasets (resricted to 64 rankers) to compare the average Condorcet regret of 100 runs of REX3 with $T = 10^5$ to the corresponding halved[1] theoretical bounds from Corollary 4.1 for various values of γ. The results of this experiment are summarized in Figure 5.1. The colored areas around the curves in all the subsequent figures show the minimal and maximal values over the runs.

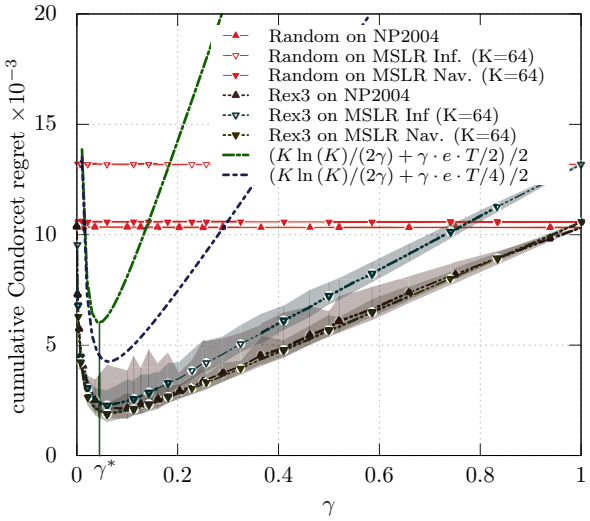


FIGURE 5.1: Empirical validation of Corollary 4.1.

[1] As mentioned in Section 2.4.3, the utility-based bandit regret is indeed twice the Condorcet regret.

We plotted two theoretical curves: one with a conservative $\mathbb{G}_{max} = T/2$, and a riskier one with $\mathbb{G}_{max} = T/4$. This experiment illustrates the dual impact of the γ parameter on the exploration/exploitation tradeoff: a low value reduces both the exploration and the reactivity of the algorithm to unexpected feedbacks and a high value tends to uniformize exploration while increasing reactivity. It also shows that the theoretical optimal γ^* we obtain with Equation (4.6) is a good guess even with a conservative upper-bound for $\mathbb{G}_{max}$.

5.2 Interleave filtering experiments

For our experiments we have considered the following state of the art algorithms: BTM by Yue and Joachims [2011] (section 2.4.3) with $\gamma = 1.1$ and $\delta = 1/T$ (explore-then-exploit setting), Condorcet-SAVAGE by Urvoy et al. [2013a] (section 2.4.4) with $\delta = 1/T$, RUCB by Zoghi et al. [2014a] (section 2.4.5) with $\alpha = 0.51$, and SPARRING coupled with EXP3 by Ailon et al. [2014] (section 2.4.11). We also took the uniform sampling strategy RANDOM as a baseline. We considered three versions of REX3 two non-anytime versions where the optimal γ^* is computed beforehand according to (4.6) with $\mathbb{G}_{max}$ set respectively to $T/2$ and $T/10$ and one anytime version where γ^* is recomputed at each time step according to Eq. (4.6) [see Seldin et al., 2012, for details about this form of "doubling trick"].

A point which makes the comparison difficult is that some algorithms are anytime while others require the horizon as input. For anytime algorithms, namely RANDOM, RUCB and REX3 with adaptive γ, we displayed the average over 100 runs of the progressive accumulation of regret while for non-anytime algorithms, namely BTM, CSAVAGE, SPARRING and other versions of REX3 we displayed the average over 50 runs of the final cumulative regret for several fixed and known horizons. This protocol is slightly favorable to non-anytime algorithms which benefit from more information. However, for elimination algorithms like BTM and CSAVAGE the difference between the anytime regret and the non-anytime regret is small. For adversarial algorithms like SPARRING and REX3 the "doubling trick" can be applied to make them anytime: the adaptive γ version of REX3 is an example of such a fixed-to-anytime transformation.

The results of these experiments are summarized in Figure 5.2 and 5.3. On regret plots, both time and regret scales are logarithmic ($\sqrt{t}$ hence appears as $t/2$).

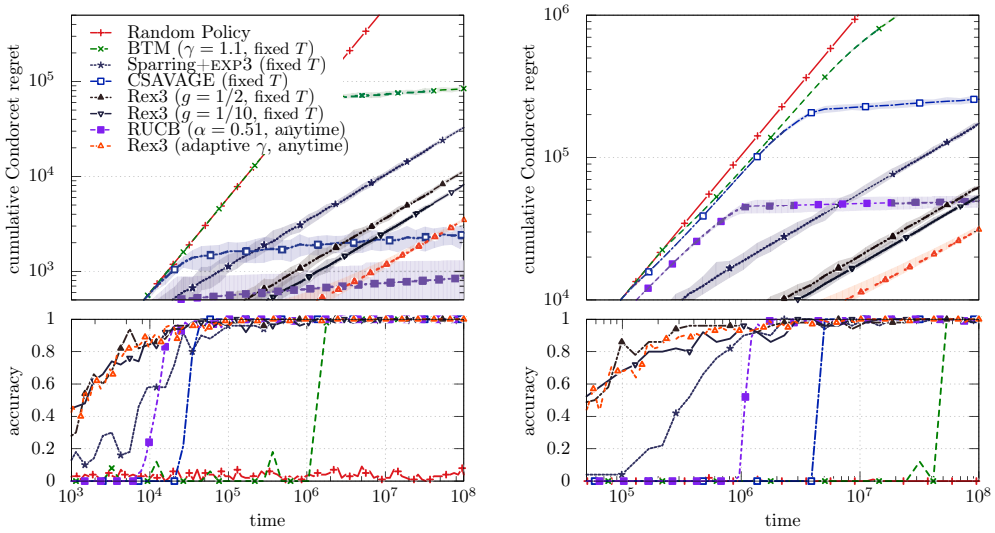


FIGURE 5.2: Regret and accuracy plots averaged over 100 runs (50 runs for fixed-horizon algorithms) respectively on ARXIV dataset (6 rankers) and LETOR NP2004 dataset (64 rankers).

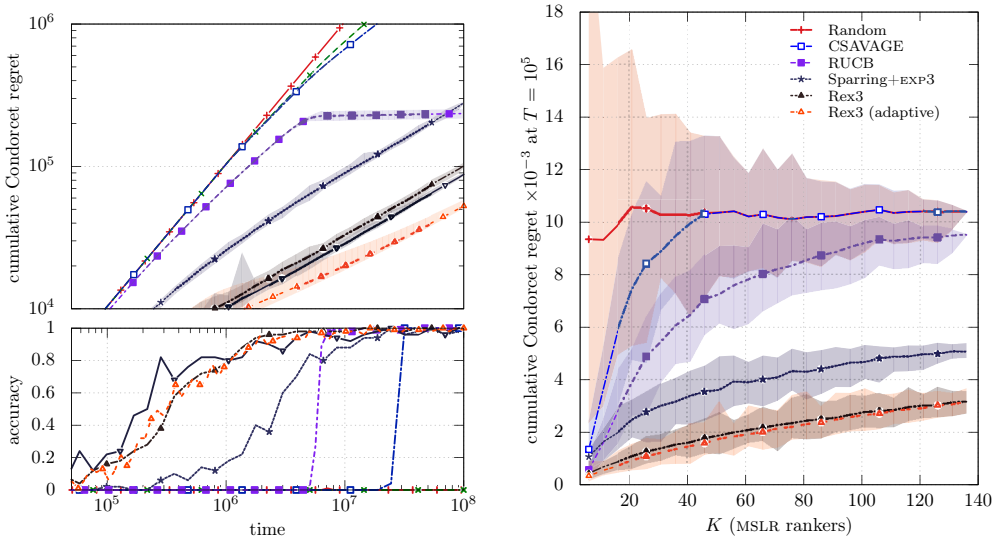


FIGURE 5.3: On the left: average regret and accuracy plots on MSLR30K with navigational queries ($K = 136$ rankers). On the right: same dataset, average regrets for a fixed $T = 10^5$ and K varying from 4 to 136.

As expected, the adversarial-setting algorithms SPARRING and REX3 follow an $\mathcal{O}(\sqrt{T})$ regret curve while the stochastic-setting algorithms follow an $\mathcal{O}(\ln T)$ curve. Among the adversarial-setting algorithms, REX3 is shown to outperform SPARRING on all datasets. In the long run, adversarial-setting algorithms continue exploring

and cannot compete in terms of regret against stochastic-setting algorithms, but the accuracy curves show that the cost of this exploration is very small. Moreover, for small horizons or high number of rankers, REX3 is extremely competitive against other algorithms. This difference is clearly illustrated on the left-hand side of Figure 5.3 where we show the evolution of the expected cumulative regret at a fixed time horizon ($T = 10^5$) according to the number of arms. To obtain this plot we averaged the regret over 50 runs. For each K and each run we sampled uniformly K dimensions of the original 136×136 MSLR30K navigational preference matrix.

Figure 5.4 gives results for smaller number of rankers on NP2004 dataset. We give the anytime runs for BTM and SAVAGE too with a conservative $\delta = 10^{-8}$. Figure 5.5 gives results for the experiments on MSLR30K dataset. There is no Condorcet winner on the left-hand-side informational queries matrix (we took a Copeland winner as a placeholder but the regret is negative for some arms).

On Figure 5.6, we added an experiment we made with Sparring coupled with UCB. We also considered two artificial matrices: SAVAGE and BVS. The 30×30 SAVAGE matrix, defined by $P_{i,j} = \frac{1}{2} + j/(2K)$ for $i < j$ as described in Urvoy et al. [2013a]. The 20×20 BVS matrix is defined by: $P_{1,j} = 0.51$ for any $j > 1$ and $P_{i,j} = 1$ for any $1 < i < j$. Its Condorcet winner has a low Borda score (9.69 against 18.49 for the Borda winner) which makes it difficult for algorithms to find the real Condorcet winner. These experiments results are summarized in Figure 5.7.

We conclude these experiments by a non-stationary utility-based dueling bandit simulation where the expected reward gap $\Delta(t)$ between the best arm and the others is set in order to deceive stochastic algorithms (see Figure 5.8). The rewards are taken from Bernoulli distributions. The best arm has a time-dependent expected reward equal to $1/2 + \Delta(t)$ with $\Delta(t) = \sqrt{K \cdot \log(t)/t}$. The others arms' rewards are stationary with a mean of $1/2$. The gap function $\Delta(t)$ has been chosen to deceive stochastic algorithms: $\mathcal{O}\left(\frac{K \cdot \log(T)}{\Delta(T)}\right) \sim \mathcal{O}\left(\sqrt{KT \cdot \log(T)}\right)$. To ease reading we provide the same plot with logarithmic scale on the left and linear scale on the right.

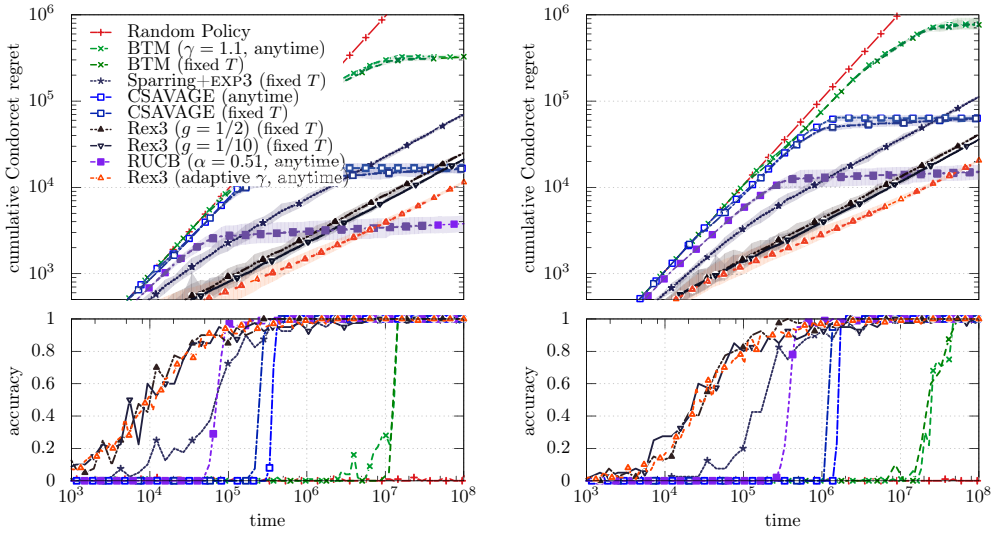


FIGURE 5.4: Average regret and accuracy plots on LETOR NP2004 with respectively 16, and 32 rankers.

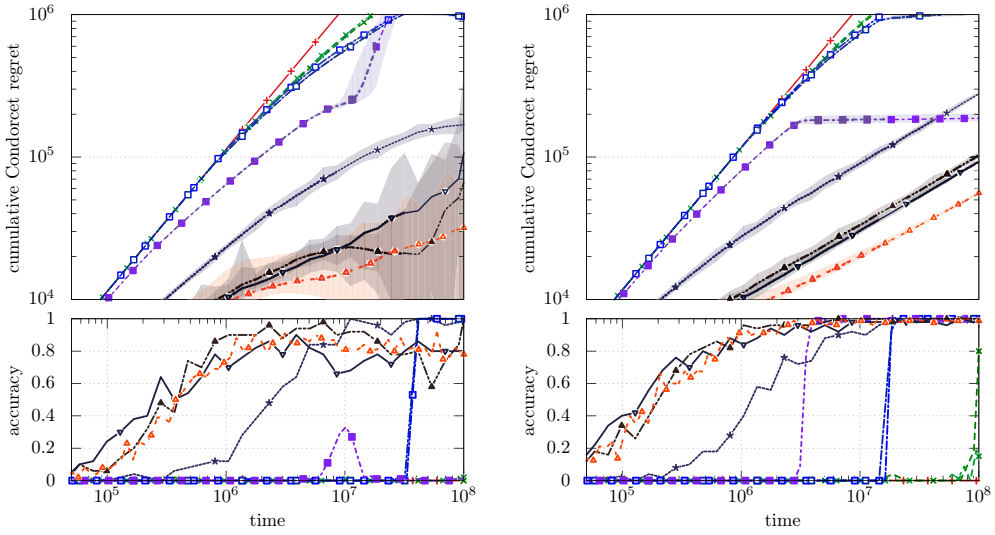


FIGURE 5.5: Expected regret and accuracy plots on MSLR30K with respectively informational and perfect navigational queries (136 rankers).

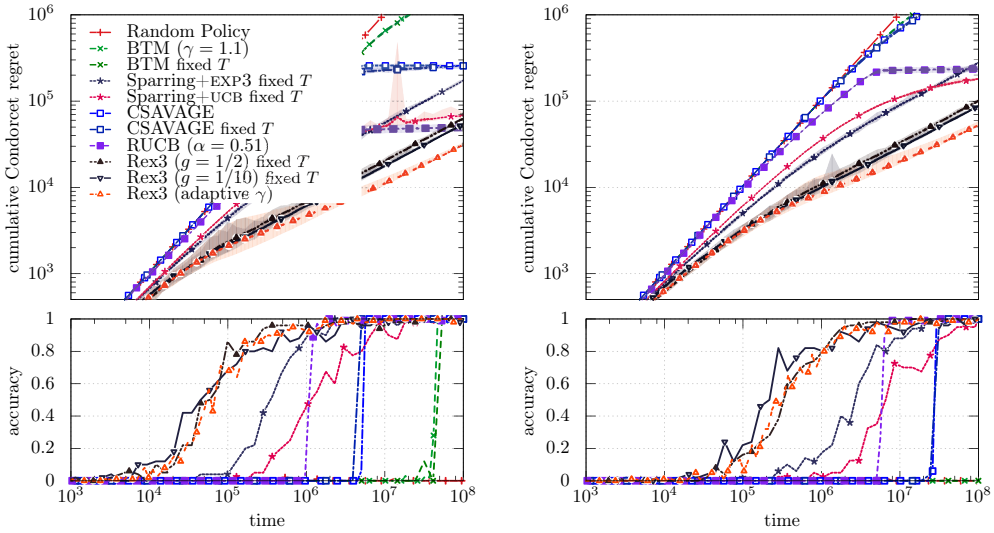


FIGURE 5.6: Average regret and accuracy plots respectively on LETOR NP2004 (64 rankers) and MSLR30K navigational queries (136 rankers) with Sparring coupled with a standard UCB MAB.

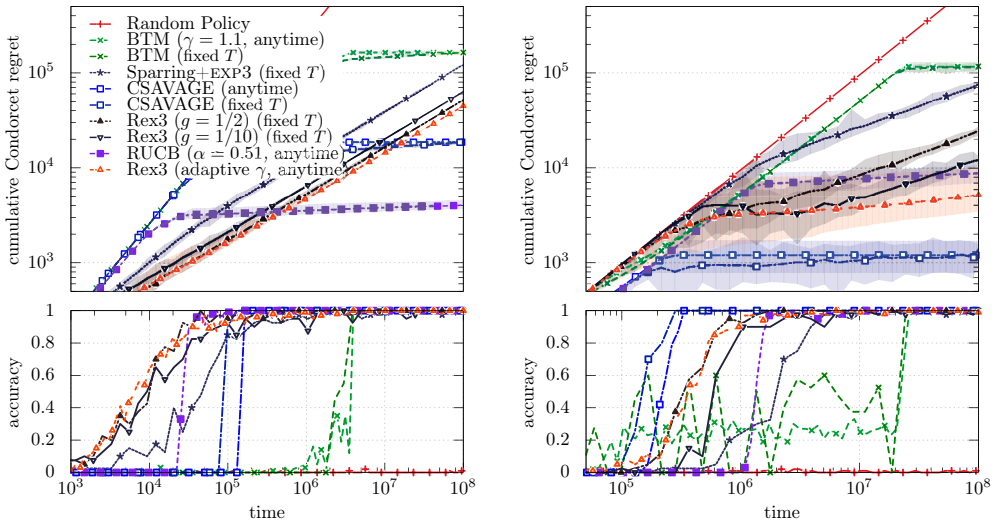


FIGURE 5.7: Average regret and accuracy plots respectively on SAVAGE and BVS artificial matrices.

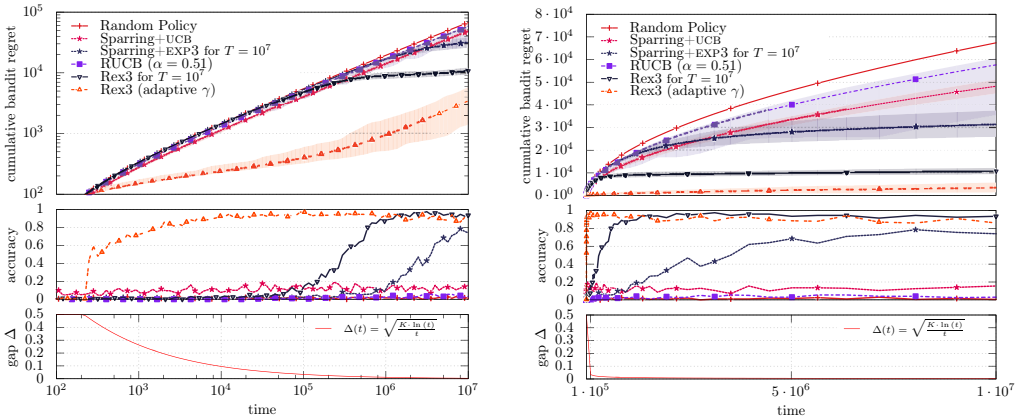


FIGURE 5.8: An experiment on a synthetic utility-based 10-armed dueling bandits problem with non-stationary rewards.

Chapter 6

Dueling Bandits as Partial Monitoring games

In this chapter, we shall see how the Bernoulli dueling bandits can be formulated as a partial monitoring game (detailed in Section 1.5). Since partial monitoring is a generic way of expressing online learning problems with incomplete feedback, formulating dueling bandits as a partial monitoring game opens a way for us to place the dueling bandits in the hierarchy of other similar learning problems.

In Section 6.1, we formalize the Bernoulli utility-based dueling bandits as a partial monitoring game. In Section 6.2, we place the dueling bandits in the partial monitoring hierarchy provided in Section 1.5.2. In Section 6.3, we investigate if the general partial monitoring algorithms can be used for the dueling bandits.

6.1 Formalization of dueling bandits as Partial Monitoring game

To recapitulate, a partial monitoring game (PM) is defined by a tuple $\langle \boldsymbol{N}, \boldsymbol{M}, \boldsymbol{\Sigma}, \mathcal{G}, \mathcal{H} \rangle$ where $\boldsymbol{N}$, $\boldsymbol{M}$, $\boldsymbol{\Sigma}$, $\mathcal{G}$ and $\mathcal{H}$ are the action set, the outcome set, the feedback alphabet, the reward function and the feedback function respectively. To each action $I \in \boldsymbol{N}$ and outcome $J \in \boldsymbol{M}$, the *reward function* $\mathcal{G}$ associates a real-valued gain $\mathcal{G}(I, J)$ and the *feedback function* $\mathcal{H}$ associates a feedback symbol $\mathcal{H}(I, J) \in \boldsymbol{\Sigma}$.

An action in the utility-based dueling bandits model consists of selecting a pair (a, b) of arms. However, symmetric actions like (a, b) and (b, a) lead to the same gains and provide equally informative feedback. Hence the action set for the learner

can be restricted to $\boldsymbol{N} = \{(a,b) : 1 \leq a, b \leq K, a \leq b\}$. The outcome set consists of environment outcomes which are arm reward vectors $\boldsymbol{m}$ where $\boldsymbol{m}_a$ is the reward for arm a. The feedback alphabet $\boldsymbol{\Sigma} = \{\square, \diamond, \blacksquare\}$ where $\square, \diamond$ and $\blacksquare$ indicate loss, tie and win for the first arm in the duo of arms selected. When the environment selects an outcome $\boldsymbol{m} \in \boldsymbol{M}$ and the learner selects a duel/action $(a,b) \in \boldsymbol{N}$, the instantaneous gain $\mathcal{G}((a,b), \boldsymbol{m})$ and feedback $\mathcal{H}((a,b), \boldsymbol{m})$ are as follows:

$$\mathcal{G}((a,b), \boldsymbol{m}) = \frac{\boldsymbol{m}_a + \boldsymbol{m}_b}{2} \qquad \mathcal{H}((a,b), \boldsymbol{m}) = \begin{cases} \square & \text{if } \boldsymbol{m}_a < \boldsymbol{m}_b \quad \textit{(loss)} \\ \diamond & \text{if } \boldsymbol{m}_a = \boldsymbol{m}_b \quad \textit{(tie)} \\ \blacksquare & \text{if } \boldsymbol{m}_a > \boldsymbol{m}_b \quad \textit{(win)} \end{cases}$$

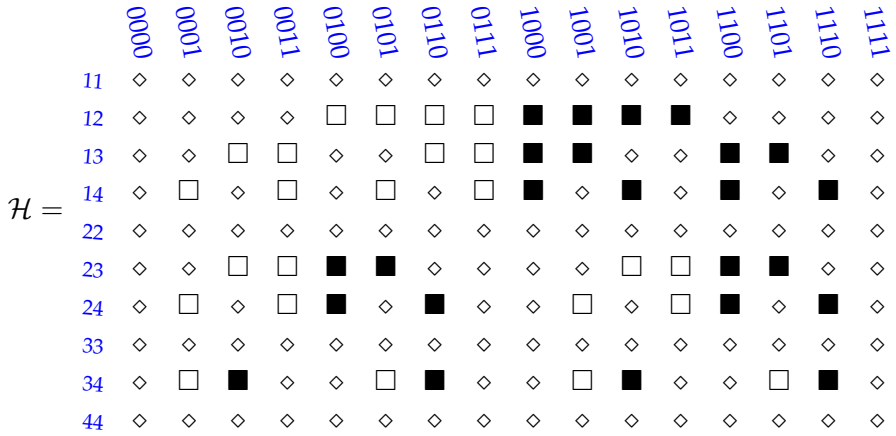
To illustrate this formalism, we encode a 4-armed binary-gain dueling bandit problem as a PM problem in Figures 6.1 and 6.2. In the provided figures, the first element of every column is of the form $\boldsymbol{m}_1\boldsymbol{m}_2\boldsymbol{m}_3\boldsymbol{m}_4$ where $\boldsymbol{m}_a$ is the gain for i^{th} arm. The first element of every row is of the form d_1d_2 where d_1 is the first arm being picked and d_2 being the second. Figure 6.3 shows the signal matrix for action (12) i.e. when arms 1 and 2 are picked. Recall from definition 1.5 that, the signal matrix for an action is the incidence matrix of symbols and outcomes.

$\mathcal{G} =$

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
11	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
12	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	1
13	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1
14	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1
22	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
23	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1
24	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1
33	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
34	0	$\frac{1}{2}$	$\frac{1}{2}$	1	0	$\frac{1}{2}$	$\frac{1}{2}$	1	0	$\frac{1}{2}$	$\frac{1}{2}$	1	0	$\frac{1}{2}$	$\frac{1}{2}$	1
44	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

FIGURE 6.1: Gain matrix $\mathcal{G}$ for a 4-armed binary dueling bandits resulting in 10 non-duplicate actions and 16 possible outcomes.

As shown above, the formulation of a K-armed Bernoulli dueling bandit problem as a partial monitoring game requires matrices of dimension $\binom{K}{2} \times 2^K$ for gain matrix and feedback matrix. Even for moderate values of K, this requirement is



$\mathcal{H} =$

FIGURE 6.2: Feedback matrix $\mathcal{H}$ for the same problem as in Figure 6.1.

$\mathcal{S}_{(12)} =$

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
□	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
◇	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
■	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

FIGURE 6.3: Signal matrix for action (12) for the same problem as in Figure 6.1.

impractical. In the next section, we see how the dueling bandits fit into the partial monitoring hieracrhy.

6.2 Dueling bandits in the partial monitoring hierarchy

This section examines the place of the dueling bandit problem in the hierarchy of partial monitoring problems described earlier. Note that the existence of the REX3 algorithm (Section 4.1) with a $\tilde{\Theta}\left(\sqrt{KT}\right)$ regret guarantee is enough to state that dueling bandit is an *easy game* according to the hierarchy described in Theorem 1.1, but our aim here is to retrieve this result from the PM machinery.

Theorem 6.1 (Duelings bandits: locally observable)**.** *In a binary utility-based dueling bandit problem with more than two arms, all the pairs of actions are locally observable.*

Proof. Consider a dueling bandit problem as defined above with binary gains and $K > 2$ arms. The signal matrix of any action $(a, b) \in \boldsymbol{N}^2$ is defined as follows:

$$S_{(a,b)}(\square, \boldsymbol{m}) = \mathbb{1}_{(\boldsymbol{m}_a < \boldsymbol{m}_b)} \quad S_{(a,b)}(\diamond, \boldsymbol{m}) = \mathbb{1}_{(\boldsymbol{m}_a = \boldsymbol{m}_b)} \quad S_{(a,b)}(\blacksquare, \boldsymbol{m}) = \mathbb{1}_{(\boldsymbol{m}_a > \boldsymbol{m}_b)}$$

Recall that, gain vector $\boldsymbol{g}_i$ denotes the column vector consisting of i^{th} row in $\mathcal{G}$. In the following, we show that for any pair of actions (a, b) and (a', b'), $\boldsymbol{g}_{(a',b')} - \boldsymbol{g}_{(a',b')}$ is locally observable. For the sake of readability, Let us consider $S^{\blacksquare}$, $S^{\diamond}$ and $S^{\square}$ to be the column vectors containing the rows pertaining to the symbols $\blacksquare$, $\diamond$ and $\square$ of the signal matrix S respectively. We consider the following two cases for the pair of actions which together cover all the possibilities:

- A pair of actions that share at-least one common arm:

  1. Actions (a, c) and (c, b). For any binary gain outcome $\boldsymbol{m}$, we have :

$$\begin{aligned}
\boldsymbol{g}_{(a,c)} - \boldsymbol{g}_{(c,b)} &= \left(\frac{\boldsymbol{m}_a + \boldsymbol{m}_k}{2} - \frac{\boldsymbol{m}_k + \boldsymbol{m}_b}{2}\right)_{\boldsymbol{m}\in\boldsymbol{M}} \\
&= 0.5\,(\boldsymbol{m}_a - \boldsymbol{m}_b)_{\boldsymbol{m}\in\boldsymbol{M}} = 0.5\left[(\boldsymbol{m}_a - \boldsymbol{m}_b)(\boldsymbol{m}_a - \boldsymbol{m}_b)^2\right]_{\boldsymbol{m}\in\boldsymbol{M}} \quad \text{(binary gains)} \\
&= 0.5\left[\frac{(\boldsymbol{m}_a - \boldsymbol{m}_b + 1)}{2}(\boldsymbol{m}_a - \boldsymbol{m}_b)^2 - \frac{(\boldsymbol{m}_b - \boldsymbol{m}_a + 1)}{2}(\boldsymbol{m}_a - \boldsymbol{m}_b)^2\right]_{\boldsymbol{m}\in\boldsymbol{M}} \\
&= 0.5\left(\mathbb{1}_{(\boldsymbol{m}_a > \boldsymbol{m}_b)} - \mathbb{1}_{(\boldsymbol{m}_b > \boldsymbol{m}_a)}\right)_{\boldsymbol{m}\in\boldsymbol{M}} = 0.5\left(S^{\blacksquare}_{(a,b)} - S^{\square}_{(a,b)}\right) \qquad (6.1)
\end{aligned}$$

     So, $\boldsymbol{g}_{(a,c)} - \boldsymbol{g}_{(c,b)}$ falls in the row space of the signal matrix of the action (a, b) and hence in the row space of the signal matrix of the neighborhood action set. (refer definition 1.6)

  2. Actions (a, c) and (b, c). Similarly, $\boldsymbol{g}_{(a,c)} - \boldsymbol{g}_{(b,c)} = 0.5S^{\blacksquare}_{(a,b)} - 0.5S^{\square}_{(a,b)}$.

- No common arm ($a \neq a' \neq b \neq b'$): In this case,

$$\begin{aligned}
\boldsymbol{g}_{(a,b)} - \boldsymbol{g}_{(a',b')} &= \boldsymbol{g}_{(a,b)} - \boldsymbol{g}_{(a,b')} + \boldsymbol{g}_{(a,b')} - \boldsymbol{g}_{(a',b')} \\
&= 0.5\left(S^{\blacksquare}_{(b,b')} - S^{\square}_{(b,b')} + S^{\blacksquare}_{(a,a')} - S^{\square}_{(a,a')}\right) \qquad \text{Using equation (6.1)}
\end{aligned}$$

Hence, for any pair of actions (a, b) and (a', b'), $\boldsymbol{g}_{(a,b)} - \boldsymbol{g}_{(a',b')}$ falls in the row space of the signal matrix of the neighborhood action set i.e. $\boldsymbol{g}_{(a,b)} - \boldsymbol{g}_{(a',b')} \in \operatorname{Im} S^{\intercal}_{((a,b)(a',b'))}$ and therefore it is locally observable. So, by extension, the binary dueling bandit problem is locally observable and hence we arrive at the following corollary. □

Corollary 6.1. *According to the hierarchy described in theorem 1.1, the binary dueling bandit problem is* easy *and its regret is* $\tilde{\Theta}(\sqrt{T})$.

6.3 Partial monitoring algorithms and their use for dueling bandits

FEEDEXP3 by Piccolboni and Schindelhauer [2001] was the first algorithm for finite partial monitoring games. For its application, there is an important pre-condition – existence of a matrix $\mathcal{B}$ such that $\mathcal{B}\mathcal{H} = \mathcal{G}$. We prove by contradiction that such a matrix $\mathcal{B}$ doesn't exist for the dueling bandit problem. Let us assume $\mathcal{B}$ exists. Therefore, for any action $(a, b) \in \boldsymbol{N}$ and any outcome vector $\boldsymbol{m} \in \boldsymbol{M}$,

$$\mathcal{G}((a,b),\boldsymbol{m}) = \sum_{i',j'=1}^{K} \mathcal{B}_{((a,b)(a',b'))} \cdot \mathcal{H}_{((a',b')(\boldsymbol{m}))}$$

Consider $\boldsymbol{m} = 0 \ldots 0$, i.e. the gain of every arm is 0. In this case, the gain of any action (a, b) is 0 and the feedback for every action is $\diamond$, therefore

$$0 = \sum_{i',j'=1}^{K} \mathcal{B}_{((a,b)(a',b'))} \cdot \diamond \tag{6.2}$$

Now consider $\boldsymbol{m} = 1 \ldots 1$, i.e. the gain of every arm is 1. In this case, the gain of any action (a, b) is 1 and feedback of every action is $\diamond$, therefore

$$1 = \sum_{i',j'=1}^{K} \mathcal{B}_{((a,b)(a',b'))} \cdot \diamond \tag{6.3}$$

Eq.(1) and eq.(2) reach a contradiction, therefore our assumption that $\mathcal{B}$ exists is incorrect. Fortunately, the authors also provide a general algorithm which performs several matrices transformations to sidestep this pre-condition. These transformations are studied thoroughly in [Bartók, 2012].

BALATON by Bartók et al. [2011], CBP-vanilla and CBP by Bartók [2012] belong to the family of algorithms for the locally observable PM games as does GLOBAL-EXP3 by Bartók [2013]. Although, for GLOBAL-EXP3, its regret bound of $\tilde{\mathcal{O}}(\sqrt{N'T})$ does not directly depend on the number of actions, but rather on the structure of games as N' is the size of the largest *point-local game*. We can however provide a counter-example for utility-based dueling bandits where $N' \approx K^2$.

We use the notations from Bartók [2013]. Consider a p in the probability simplex

TABLE 6.1: Summary of partial monitoring algorithms

Algorithm	Setting	Optimality	Regret
FEEDEXP3 [1]	Adversarial	Not in T or N	$\tilde{\mathcal{O}}(T^{2/3}K)$
BALATON [2]	Stochastic	Not in T or N	$\tilde{\mathcal{O}}(K\sqrt{T})$
CBP [3]	Stochastic	in T, not in N	$\tilde{\mathcal{O}}(K^2 logT)$
GLOBAL-EXP3 [4]	Adversarial	in T, not in N	$\tilde{\mathcal{O}}(K\sqrt{T})$
SAVAGE [5]	Stochastic	in T, not in N	$\mathcal{O}(K^2 logT)$
NEIGHBORHOOD WATCH [6]	Adversarial	in T, not in N	$\tilde{\mathcal{O}}(K\sqrt{T})$
REX3 [7]	Adversarial	in T and N	$\tilde{\mathcal{O}}(\sqrt{KT})$

$\Delta_{|M|}$ where all the arms have maximal gains. For this p, all the actions are optimal therefore this point belongs to all the cells in the cell-decomposition. Hence, according to definition 6 in Bartók [2013], there exists a point-local game consisting of all the $K(K+1)/2$ non-duplicate actions. Therefore the upper bound of GLOBALEXP3 translates to $\tilde{O}(K\sqrt{T})$ for utility-based dueling bandits.

The table 6.1 summarizes the salient features of these PM algorithms. We can clearly see that none of them, except REX3 is optimal with respect to the number of actions N. Please note that for the dueling bandits problem, $N \approx K^2$.

This brings us to the end of this chapter. Next in Appendix A, we shall see the detailed proof of Theorem 4.1.

[1] (Piccolboni and Schindelhauer [2001])
[2] (Bartók et al. [2011])
[3] (Bartók [2012])
[4] (Bartók [2013])
[5] (Urvoy et al. [2013b])
[6] (Foster and Rakhlin [2011])
[7] (Gajane et al. [2015])

Appendix A

Detailed Proof of Theorem 4.1

For better readabilty, we simply write a, b instead of a_t, b_t when referring of the arms chosen by the algorithm. We also frequently drop the time indices for $p_\cdot(t)$ and $x_\cdot(t)$.

Proof. Let $W_t = w_1(t) + w_2(t) + \cdots + w_K(t)$

$$\frac{W_{t+1}}{W_t} = \sum_{i=1}^{K} \frac{w_i(t+1)}{W_t} = \sum_{i=1}^{K} \frac{w_i(t)}{W_t} e^{(\gamma/K)\hat{c}_i(t)}$$

By substituting the values of $\frac{w_i(t)}{W_t}$, we get:

$$\frac{W_{t+1}}{W_t} = \sum_{i=1}^{K} \frac{p_i(t) - \gamma/K}{1-\gamma} e^{(\gamma/K)\hat{c}_i(t)}$$

Using the inequality $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$, we get:

$$\begin{aligned}
\frac{W_{t+1}}{W_t} &\leq \sum_{i=1}^{K} \frac{p_i(t) - \gamma/K}{1-\gamma} \left(1 + (\gamma/K)\hat{c}_i(t)\right) + \sum_{i=1}^{K} \frac{p_i(t) - \gamma/K}{1-\gamma} \left((e-2)(\gamma^2/K^2)\hat{c}_i(t)^2\right) \\
&\leq \underbrace{\sum_{i=1}^{K} \frac{p_i(t) - \gamma/K}{1-\gamma}}_{=1} + \frac{\gamma/K}{1-\gamma} \left(\underbrace{\sum_{i=1}^{K} p_i(t)\hat{c}_i(t)}_{=0 \text{ see (A.2)}} - \frac{\gamma}{K} \sum_{i=1}^{K} \hat{c}_i(t) \right) + \frac{(e-2)(\gamma^2/K^2)}{1-\gamma} \sum_{i=1}^{K} p_i(t)\hat{c}_i(t)^2 \\
&\leq 1 + \frac{\gamma/K}{1-\gamma} \left(0 - \frac{\gamma}{K} \sum_{i=1}^{K} \hat{c}_i(t) \right) + \frac{(e-2)(\gamma^2/K^2)}{1-\gamma} \sum_{i=1}^{K} p_i(t)\hat{c}_i(t)^2 \\
&\leq 1 - \frac{\gamma^2/K}{1-\gamma} \left(\underbrace{\frac{1}{K} \sum_{i=1}^{K} \hat{c}_i(t)}_{=-M_1} \right) + \frac{(e-2)(\gamma^2/K)}{1-\gamma} \underbrace{\frac{1}{K} \sum_{i=1}^{K} p_i(t)\hat{c}_i(t)^2}_{=M_2} \qquad \text{(A.1)}
\end{aligned}$$

$$\sum_{i=1}^{K} p_i(t)\hat{c}_i(t) = \sum_{i=1}^{K} p_i(t)\left(\mathbb{1}_{(i=a)}\frac{x_a - x_b}{2p_a}\right) + \sum_{i=1}^{K} p_i(t)\left(\mathbb{1}_{(i=b)}\frac{x_b - x_a}{2p_b}\right)$$
$$= \frac{x_a - x_b}{2} + \frac{x_b - x_a}{2} = 0 \tag{A.2}$$

From (A.1) and (A.2), we obtain:

$$\frac{W_{t+1}}{W_t} \leq 1 + \frac{\gamma^2/K}{1-\gamma}M_1 + \frac{(e-2)(\gamma^2/K)}{1-\gamma}M_2$$

Taking logarithms and using the inequality $1 + x \leq e^x$

$$\ln\frac{W_{t+1}}{W_t} \leq \frac{\gamma^2/K}{1-\gamma}M_1 + \frac{(e-2)(\gamma^2/K)}{1-\gamma}M_2$$

Summing over t, we get:

$$\ln\frac{W_{T+1}}{W_1} \leq \frac{\gamma^2/K}{1-\gamma}M_1 + \frac{(e-2)(\gamma^2/K)}{1-\gamma}M_2 \tag{A.3}$$

For any arm j we have:

$$\sum_{t=1}^{T}\frac{\gamma}{K}\hat{c}_j(t) - \ln(K) \leq \ln\frac{W_{T+1}}{W_1} \tag{A.4}$$

The proof of the above inequality is given in appendix A.1. By combining (A.3) and (A.4) , we get:

$$\sum_{t=1}^{T}\frac{\gamma}{K}\hat{c}_j(t) - \ln(K) \leq \frac{\gamma^2/K}{1-\gamma}M_1 + \frac{(e-2)\gamma^2/K}{1-\gamma}M_2 \tag{A.5}$$

Taking the expectation over the algorithm's randomization, we obtain:

$$\sum_{t=1}^{T}\frac{\gamma}{K}\underbrace{\mathbb{E}_{\sim p}\hat{c}_j(t)}_{\text{Lemma 4.1}} - \ln(K) \leq \frac{\gamma^2/K}{1-\gamma}\sum_{i=t}^{T}\underbrace{\mathbb{E}_{\sim p}M_1}_{\text{see (A.7)}} + \frac{(e-2)(\gamma^2/K)}{1-\gamma}\sum_{i=t}^{T}\underbrace{\mathbb{E}_{\sim p}M_2}_{\text{see (A.8)}}$$

From Lemma 4.1 which proof is detailed in appendix A.2, we have:

$$\mathbb{E}_{\sim p}\hat{c}_j(t) = x_j - \mathbb{E}_{\sim p}(x_a) \tag{A.6}$$

By averaging (A.6) over the arms, we obtain:

$$\mathbb{E}_{\sim p(t)} M_1 = \mathbb{E}_{\sim p}\left(-\frac{1}{K}\sum_{i=1}^{K}\hat{c}_i(t)\right) = -\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{\sim p}\hat{c}_i(t) = \mathbb{E}(x_a) - \frac{1}{K}\sum_{i=1}^{K} x_i \quad \text{(A.7)}$$

The following result is detailled in appendix A.3:

$$\begin{aligned}\mathbb{E}_{\sim p(t)} M_2 &= \mathbb{E}_{\sim p(t)}\left(\frac{(p_a+p_b)(x_a-x_b)^2}{4Kp_ap_b}\right) = \frac{1}{2}\mathbb{E}(x_a^2) - \mathbb{E}(x_a)\frac{1}{K}\sum_{i=1}^{K} x_i + \frac{1}{2K}\sum_{i=1}^{K} x_i^2 \\ &\leq \frac{1}{2}\mathbb{E}(x_a) - \mathbb{E}(x_a)\frac{1}{K}\sum_{i=1}^{K} x_i + \frac{1}{2K}\sum_{i=1}^{K} x_i \qquad \text{as } \forall i,\ x_i \in [0,1] \quad \text{(A.8)}\end{aligned}$$

From Lemma 4.1, (A.7), and (A.8), we get for any j:

$$\begin{aligned}&\frac{\gamma}{K}\left(\sum_{t=1}^{T} x_j - \sum_{t=1}^{T}\mathbb{E}(x_a)\right) - \ln(K) \leq \frac{\gamma^2/K}{1-\gamma}\sum_{i=t}^{T}\left(\mathbb{E}(x_a) - \frac{1}{K}\sum_{i=1}^{K} x_i\right) \\ &+ \frac{(e-2)\gamma^2/K}{2(1-\gamma)}\sum_{i=t}^{T}\left(\mathbb{E}(x_a^2) - 2\mathbb{E}(x_a)\frac{1}{K}\sum_{i=1}^{K} x_i + \frac{1}{K}\sum_{i=1}^{K} x_i^2\right) \quad \text{(A.9)}\end{aligned}$$

By definition, $\mathbb{G}_{max} = \max_j \sum_{t=1}^{T} x_j$, $\mathbb{EG}_{alg} = \sum_{t=1}^{T}\mathbb{E}_{\sim p(t)}(x_a)$, and $\mathbb{EG}_{unif} = \sum_{t=1}^{T}\frac{1}{K}\sum_{i=1}^{K} x_i$. We can hence rewrite Equation (A.9) into:

$$\begin{aligned}\mathbb{G}_{max} - \mathbb{EG}_{alg} - \frac{K\ln K}{\gamma} &\leq \frac{\gamma}{1-\gamma}\left(\mathbb{EG}_{alg} - \mathbb{EG}_{unif}\right) \\ &+ \frac{(e-2)\gamma}{2(1-\gamma)}\left(\mathbb{EG}_{alg} + \mathbb{EG}_{unif} - 2\sum_{t=1}^{T}\sum_{i=1}^{K}\frac{x_i}{K}\mathbb{E}(x_a)\right)\end{aligned}$$

Let ε be such that $\forall i, t\ \varepsilon \leq x_i(t)$ then:

$$\mathbb{G}_{max} - \mathbb{EG}_{alg} \leq \frac{K\ln K}{\gamma} + \frac{e\gamma}{2(1-\gamma)}\mathbb{EG}_{alg} - \frac{(4-e+(e-2)\varepsilon)\gamma}{2(1-\gamma)}\mathbb{EG}_{unif}$$

Assuming $\gamma \leq \frac{1}{2}$:

$$\mathbb{G}_{max} - \mathbb{EG}_{alg} \leq \frac{K\ln K}{\gamma} + \gamma\left[e\mathbb{EG}_{alg} - (4-e+(e-2)\varepsilon)\,\mathbb{EG}_{unif}\right]$$

□

A.1 Proof of eq. (A.4)

For any j we have:

$$W_{T+1} = \sum_{i=1}^{K} w_i(T+1) \geq w_j(T+1).$$

Hence:

$$\begin{aligned} W_{T+1} &\geq w_j(T) e^{(\gamma/K)(\hat{c}_j(T))} \\ &= w_j(T-1) e^{(\gamma/K)(\hat{c}_j(T-1))} e^{(\gamma/K)(\hat{c}_j(T))} \\ &= w_j(1) \prod_{t=1}^{T} e^{(\gamma/K)(\hat{c}_j(t))} \end{aligned}$$

and

$$\ln W_{T+1} \geq \ln w_j(1) + \sum_{t=1}^{T} \frac{\gamma}{K} \hat{c}_j(t)$$

Since $w_j(1) = 1$ for any j, it turns out that: $\sum_{t=1}^{T} \frac{\gamma}{K} \hat{c}_j(t) - \ln(K) \leq \ln(\frac{W_{T+1}}{W_1})$ □

A.2 Proof of Lemma 4.1

$$\begin{aligned} \hat{c}_i(t) &= \mathbb{1}_{(i=a_t)} \frac{\psi(x_{a_t} - x_{b_t})}{2p_{a_t}(t)} + \mathbb{1}_{(i=b_t)} \frac{\psi(x_{b_t} - x_{a_t})}{2p_{b_t}(t)} \\ \mathbb{E}_{(a,b)\sim p(t)} \hat{c}_i(t) &= \sum_{j=1}^{K} \sum_{k=1}^{K} p_j(t) p_k(t) \left(\mathbb{1}_{(i=j)} \frac{\psi(x_j - x_k)}{2p_j} + \mathbb{1}_{i=k)} \frac{\psi(x_k - x_j)}{2p_k} \right) \\ &= \sum_{j=1}^{K} \sum_{k=1}^{K} p_j p_k \mathbb{1}_{(i=j)} \frac{\psi(x_j - x_k)}{2p_j} + \sum_{j=1}^{K} \sum_{k=1}^{K} p_j p_k \mathbb{1}_{(i=k)} \frac{\psi(x_k - x_j)}{2p_k} \\ &= \frac{1}{2} \sum_{k=1}^{K} p_k \psi(x_i - x_k) + \frac{1}{2} \sum_{j=1}^{K} p_j \psi(x_i - x_j) \\ &= \mathbb{E}_{a\sim p} \psi(x_i - x_a) \end{aligned}$$

If ψ is identity, it simplifies into:

$$\mathbb{E}_{(a,b)\sim p(t)} \hat{c}_i(t) = x_i - \mathbb{E}_{a\sim p(t)} x_a$$

□

A.3 Proof of eq. (4.5) and (A.8)

$$
\begin{aligned}
M_2 &= \frac{1}{K}\sum_{i=1}^{K} p_i(t)\hat{c}_i(t)^2 \\
&= \frac{1}{K}\sum_{i=1}^{K} p_i \left(\mathbb{1}_{(i=a)}\frac{x_a - x_b}{2p_a} + \mathbb{1}_{(i=b)}\frac{x_b - x_a}{2p_b}\right)^2 \\
&= \frac{1}{K}\sum_{i=1}^{K}\Bigg(p_i\mathbb{1}_{(i=a)}\frac{(x_a - x_b)^2}{4p_a^2} + p_i(t)\mathbb{1}_{(i=b)}\frac{(x_b - x_a)^2}{4p_b^2} \\
&\qquad \underbrace{+ 2p_i\mathbb{1}_{(i=a)}\mathbb{1}_{(i=b)}\frac{x_a - x_b}{2p_a}\frac{x_b - x_a}{2p_b}}_{=0}\Bigg) \\
&= \sum_{i=1}^{K} p_i\mathbb{1}_{(i=a)}\frac{(x_a - x_b)^2}{4Kp_a^2} + \sum_{i=1}^{K} p_i\mathbb{1}_{(i=b)}\frac{(x_b - x_a)^2}{4Kp_b^2} \\
&= \frac{(x_a - x_b)^2}{4Kp_a} + \frac{(x_b - x_a)^2}{4Kp_b} \\
&= \frac{(p_a + p_b)(x_a - x_b)^2}{4Kp_ap_b}
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}_{(a,b)\sim p(t)} M_2 &= \mathbb{E}_{(a,b)\sim p(t)}\left(\frac{(p_a + p_b)(x_a - x_b)^2}{4Kp_ap_b}\right) \\
&= \sum_{i=1}^{K}\sum_{j=1}^{K} p_ip_j\frac{(p_i + p_j)(x_i - x_j)^2}{4Kp_ip_j} \\
&= \quad \frac{1}{4K}\sum_{i=1}^{K}\sum_{j=1}^{K}(p_i + p_j)(x_i - x_j)^2 \\
&= \frac{1}{4K}\sum_{i=1}^{K}\sum_{j=1}^{K}(p_i + p_j)(x_i^2 - 2x_ix_j + x_j^2) \\
&= \frac{1}{4K}\Bigg(\sum_{i=1}^{K}\sum_{j=1}^{K} p_ix_i^2 - 2\sum_{i=1}^{K}\sum_{j=1}^{K} p_ix_ix_j + \sum_{i=1}^{K}\sum_{j=1}^{K} p_ix_j^2 \\
&\qquad + \sum_{i=1}^{K}\sum_{j=1}^{K} p_jx_i^2 - 2\sum_{i=1}^{K}\sum_{j=1}^{K} p_jx_ix_j + \sum_{i=1}^{K}\sum_{j=1}^{K} p_jx_j^2\Bigg)
\end{aligned}
$$

$$= \frac{1}{4K}\left(2K\mathbb{E}(x_a^2) - 4\mathbb{E}(x_a)\sum_{i=1}^{K} x_i + 2\sum_{i=1}^{K} x_i^2\right)$$
$$= \frac{1}{2}\left(\mathbb{E}_{a\sim p(t)}(x_a^2) - 2\mathbb{E}_{a\sim p(t)}(x_a)\frac{1}{K}\sum_{i=1}^{K} x_i + \frac{1}{K}\sum_{i=1}^{K} x_i^2\right)$$

For any $x \in [0,1]$, $x^2 < x$, hence:

$$\mathbb{E}_{(a,b)\sim p(t)} M_2 \leq \frac{1}{2}\left(\mathbb{E}_{a\sim p(t)}(x_a) - 2\mathbb{E}_{a\sim p(t)}(x_a)\frac{1}{K}\sum_{i=1}^{K} x_i + \frac{1}{K}\sum_{i=1}^{K} x_i\right)$$

□

Part III

Corrupt Bandits

Chapter 7

The Corrupt Bandits problem

In the previous part, we considered a form of unconventional feedback called relative feedback in which the learner selects two arms and receives the preference of one over the other. In this chapter, we consider another form of unconventional feedback. In this setting, the learner selects a single arm at each time period akin to the classical MAB problem. However the learner does not see the reward of the chosen arm but receives a *corrupt feedback* derived from the corresponding reward. In this chapter, we introduce the MAB problem with corrupted feedback or the corrupt bandit problem.

In Section 7.1, we provide the motivation in the form of practical applications for the MAB problem with corrupted feedback. We formally define the problem and its different variations in Section 7.2. We enlist, in Section 7.3, our contributions to the problem. In Section 7.4, we provide an overview of the prvious related work.

7.1 Motivation

Consider an organization which has a pool of advertisements to be displayed on a website. Since the space available to display them is limited, only a few of the advertisements can be used at any time. Naturally the concerned organization wants to select the set which perform the best. One of the most important indicators about the performance of an online advertisement is the number of users which clicked on it over a given period of time.

To formulate this in terms of a MAB problem, we can assume individual advertisements to be the arms. When an arm is pulled, the corresponding advertisement

is chosen to be displayed for a scheduled period of time. If a user clicks on the advertisement, it is considered a positive reward and no click constitutes a negative reward. However, the feedback is usually given only for positive rewards since propagating feedback for negative rewards as well is costly in terms of network load. The reception of a click feedback can be safely interpreted as a positive reward, but the absence of such a click (timeout) might either be a consequence of a negative reward (the user did not click on the advertisement) or the consequence of a bug or a packet loss.

Let us now briefly examine the task of adaptive routing. In this task, the learner has to select a routing path in a network with unknown delays varying over time. If the intended message is correctly transmitted to the destination along the channel selected by the learner, it constitutes a positive reward for the learner. If the selected channel is impassable, the intended message can not arrive at the destination and this event corresponds to a negative reward for the learner. In the former case, positive feedback is sent to the learner. Hence, positive feedback means the corresponding path is usable but no feedback could either mean that the corresponding channel is unusable or the feedback was dropped due to extraneous issues. If we were to model this scenario as a MAB problem, we would run into the same difficulty mentioned in the previous paragraph; namely, the feedback is not equal to the corresponding reward.

Consider the task of online recommendation. The goal of an online recommendation system is to provide recommendations to the users for items that are likely to interest them. For instance, movie streaming websites recommend a number of movies to their users. These recommendations are based on the ratings given for the items by the users, user demographics, items characteristics etc. For the sake of simplicity, let us assume only the binary ratings given by the user are employed to determine the items to be recommended to that user. Let us attempt to formulate this as a MAB problem. The items are to be considered as the arms of the bandit problem. When the learner picks an arm, the corresponding item is recommended to the user. If the user likes the recommended item, it constitutes a positive reward

for the learner and a negative reward corresponds to the case where user dislikes the recommended item. The user is then expected to give true feedback equivalent to the reward i.e. positive feedback for positive reward and vice versa. However the users might be hesitant to provide true feedback due to privacy concerns. One of the ways that the user privacy is breached is when the recommender systems share user data with third parties (Bennett and Lanning [2007]). Therefore some users are willing to surrender the benefits of useful recommendations in order to protect their privacy (Culnan [2000]). This phenomenon of personalization to privacy trade-off is described in Awad and Krishnan [2006], Chellappa and Sin [2005].

In the related task of survey systems, individuals are given a questionnaire and they are expected to provide honest responses to them. However, Warner [1965] note that some respondents attempt to evade certain questions for the reasons of modesty, fear of being thought bigoted, or merely a reluctance to confide secrets to strangers. These non-cooperative respondents either refuse outright to be surveyed or consent to be surveyed but purposefully provide wrong answers. To elicit responses from such individuals Warner [1965] propose the *randomized response method (RR)* as a survey technique to reduce potential bias due to non-response and social desirability when asking questions about sensitive behaviors and beliefs. This method asks respondents to employ randomization say with a flip of coin having bias p, the outcome of which is not available to the interviewer. Before responding to an question, the respondent flips the coin. If it is heads, the respondent answers truthfully otherwise inaccurately. By introducing random noise, the method conceals individual responses and protects respondent privacy. The randomized response method can be used to protect user privacy in online recommendation as well. If recommendation systems that allow users to employ randomized response method are to be modeled as a MAB problem, we again face the oft-mentioned complication in this section: untrustworthy feedback.

The common theme among all of the above cases is that the feedback the learner receives is not equal to the reward of the chosen arm but at the same time the feedback and the corresponding reward are nonetheless related. This presents a hindrance if we were to model these applications as a classical MAB problem as it assumes that the feedback is equal to the reward i.e. the learner sees the reward of the selected arm. This motivates us to devise a variation of the MAB problem in which the learner is able to learn from the feedback derived from, but not necessarily equal to the corresponding reward. We call this model as the MAB problem with corrupted feedback or corrupt bandits and it is formally presented in the next section.

7.2 Formalization

A corrupt bandit problem $\boldsymbol{\nu}$ is formally characterized by a set of arms $A = \{1, \ldots, K\}$ on which are indexed a list of known *corruption functions* $\{g_a\}_{a \in A}$. At each time period t, every arm $a \in A$ is associated with a reward $x_a(t)$ and a feedback $y_a(t)$. The reward vector $\mathbf{x}_t$ consists of $\{x_1(t), \ldots, x_K(t)\}$ and the feedback vector $\mathbf{y}_t$ consists of $\{y_1(t), \ldots, y_K(t)\}$. If the learner pulls the arm a at time t, it receives the corresponding (hidden) reward $x_a(t)$ and observes the feedback $y_a(t)$. We assume that, for each arm $a \in A$, there exists a loose link between the reward and the feedback through the corruption function g_a. A corrupt bandit problem can also be expressed as an instance of the partial monitoring problem introduced in Section 1.5 and we discuss this further is Chapter 12. Like the classical MAB problem (Section 1.2.1) and the dueling bandit problem (Section 2.2), the corrupt bandit problem too can be formalized in two different settings.

7.2.1 Stochastic setting

Similar to the corresponding stochastic setting in other variations of the MAB problem, the defining feature of this setting is the stationarity of the distributions associated with the arms. A stochastic corrupt bandit problem $\boldsymbol{\nu}$ is characterized by a set

of arms $A = \{1, \dots, K\}$ on which are indexed a list of unknown sub-Gaussian reward distributions $\{\nu_a\}_{a \in A}$, a list of unknown sub-Gaussian feedback distributions $\{\varsigma_a\}_{a \in A}$, and a list of known mean-corruption functions $\{g_a\}_{a \in A}$.

If the learner pulls an arm $a \in A$ at time t, it receives a reward $R_t = x_a(t)$ drawn from the distribution ν_a with mean $\mu_a^{\boldsymbol{\nu}}$ and observes a feedback $F_t = y_a(t)$ drawn from the distribution ς_a with mean $\lambda_a^{\boldsymbol{\nu}}$. We assume that a known mean-corruption function (or simply, corruption function) g_a maps the mean of the reward distribution to the mean of the feedback distribution :

$$g_a(\mu_a^{\boldsymbol{\nu}}) = \lambda_a^{\boldsymbol{\nu}}, \qquad \forall a \in A \tag{7.1}$$

Stochastic corrupt bandit problem

The corruption functions $g_1, \dots, g_K$ are revealed to the learner.
At $t \leftarrow 1, \dots, T$

1. The environment draws a reward vector $\mathbf{x}_t$ according to $\nu_1 \times \cdots \times \nu_K$ with means $\mu_1, \dots, \mu_k$ respectively.
2. The learner selects an arm $a_t \in A \coloneqq \{1, \dots, K\}$.
3. The learner receives the reward $x_{a_t}(t)$.
4. The learner observes the feedback $y_{a_t}(t)$ drawn from a Bernoulli distribution with mean $g_{a_t}(\mu_{a_t})$.

Note that these g_a functions may be completely different from one arm to another. For Bernoulli distributions, μ_a and λ_a are in $[0, 1]$ for all $a \in A$ and we assume all the corruption functions $\{g_a\}_{a \in A}$ to be continuous at-least in this interval.

Another way to define the link between the reward and the feedback is to provide a *corruption scheme* operator $\tilde{g}_a$ which maps the reward outcomes into feedback distributions. If the mean is a sufficient statistic of the reward distribution, then the learner can build its own corruption function from the corruption scheme and the two definitions are equivalent. This equivalence is true for Bernoulli distributions where most of our results apply.

7.2.2 Adversarial setting

In the adversarial setting the reward and the feedback are not assumed to be drawn from a distribution, unlike the stochastic setting. At each time period, an adversary assigns a reward value and a feedback value to every arm. Learning any information about the hidden reward from the observed feedback is not achievable if the adversary is not placed under any constraint with respect to its ability to generate the feedbacks. Hence the adversary is forced to comply by some constraints. In a similar setting proposed by Feige et al. [2015], the adversary is only allowed to corrupt to a certain number of values. In our setting, the constraint is on the average reward and the average feedback. Let $\hat{x}_a(t)$ and and $\hat{y}_a(t)$ denote the average reward and the average feedback for arm a till time t. The adversary is then restricted to set the reward and feedback values for any arm a such that the respective average feedback is derivable from the average reward with the use of a known corruption function. Overloading the symbol g, this restriction can be expressed as[1]

$$\hat{y}_a(T) \overset{con}{=} g_a(\hat{x}_a(T)) \quad \forall a \in A$$

Adversarial corrupt bandit problem

The corruption constraints $g_1, \ldots, g_K$ are revealed to the learner.
The adversary draws reward vectors $\mathbf{x}_t \in [0,1]^K$ for $t \leftarrow 1, \ldots, T$.
At $t \leftarrow 1, \ldots, T$

1. The learner selects an arm $a_t \in A \coloneqq \{1, \ldots, K\}$.
2. The learner receives the reward $x_{a_t}(t)$.
3. The learner observes the feedback $y_{a_t}(t)$ set by the adversary constraint to the condition that $\hat{y}_a(T) \overset{con}{=} g_a(\hat{x}_a(T))$

[1] $\overset{con}{=}$ is used to denote an equality constraint.

7.2.3 Randomized response as a corrupt bandit problem

The randomized response method (Warner [1965]) was briefly described in the section 7.1. It can also be used to corrupt the rewards in a MAB problem. Here we illustrate how it can be formulated as a corrupt bandit problem.

Consider a bandit problem with binary rewards and feedbacks. For each arm a, every possible reward is corrupted to one of the possible values of feedback with a certain probability given by

$$
\begin{aligned}
p_{00}(a) &\coloneqq \text{ the probability that feedback is } 0 \text{ given the reward for arm } a \text{ is } 0 \\
p_{01}(a) &\coloneqq \text{ the probability that feedback is } 0 \text{ given the reward for arm } a \text{ is } 1 \\
p_{10}(a) &\coloneqq \text{ the probability that feedback is } 1 \text{ given the reward for arm } a \text{ is } 0 \\
p_{11}(a) &\coloneqq \text{ the probability that feedback is } 1 \text{ given the reward for arm } a \text{ is } 1
\end{aligned}
$$

Hence the corruption function is $g_a : \lambda_a \coloneqq p_{10}(a) + (p_{11}(a) - p_{10}(aa))\mu_a$. The corresponding corruption scheme $\tilde{g}_a$ can be encoded by the matrix:

$$
\mathbb{M}_a \coloneqq \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} p_{00}(a) & p_{01}(a) \\ p_{10}(a) & p_{11}(a) \end{bmatrix} \end{array}
$$

The matrix $\mathbb{M}_a$ contains elements denoting the probability with which the learner sees, for arm a, the feedback given by the row index on receiving the reward given by the column index i.e.

$$
\mathbb{M}_a(x, y) \coloneqq \mathbb{P}(\text{feedback for arm a} = x \mid \text{reward for arm a} = y)
$$

Having formalized the problem in this section, we briefly enlist our contributions to the literature.

7.3 Our contributions

We introduce the setting of corrupt bandits to the MAB literature. We propose algorithms for the stochastic corrupt bandit problem [2] for exploration-exploitation as well as best arm identification. We provide the upper bounds on the regret of the introduced algorithms for exploration-exploitation and the upper bounds on the sample complexity for the introduced algorithms for best arm identification. We also provide the lower bounds to verify how our algorithms fare as compared to the best achievable performance given by the respective lower bounds. Furthermore, we describe how corrupt bandits can be used to enforce differential privacy. Our experiments show the performance of the proposed algorithms on simulated examples. We also formulate Bernoulli corrupt bandits as an instance of the partial monitoring problem.

Before studying our contributions in detail, we shall first take a look at the related work for this problem, in the next section.

7.4 Related work

To the best of our knowledge, the exact setting of corrupt bandits has not been studied previously, however it can be considered as a form of learning with incomplete feedback, therefore we shall take a look at the relevant previous work in this broad topic. Feedback could be considered as incomplete because:

I. it is provided only for a subset of instances or

II. it is provided erroneously for some/all instances

Firstly, we study scenarios in which feedback is provided only for a subset of instances belonging to particular class.

7.4.1 Positive and unlabeled learning

One of the tasks in text learning applications is to classify user-generated text according to the interest. In this task, positive examples can be found and unlabeled

[2] Henceforth, we mention the stochastic corrupt bandit problem simply as the corrupt bandit problem.

examples are abundant. In the classification literature, such an *asymmetric feedback* is called *Positive and Unlabeled (PUN)* feedback.

Denis [1998] introduce the Probably Approximately Correct (PAC) learning model from positive and unlabeled examples. They show that the decision functions are learnable in this model which contain ostensibly less information than the model with positive and negative examples as well. Denis et al. [2005] design a decision tree induction algorithm which uses only positive and unlabeled examples. Lee and Liu [2003a] transform the problem of learning from only positive and unlabeled examples into a problem of learning with noise by labeling all unlabeled examples as negative. They obtain a linear function to learn from these noisy examples while performing weighted logical regression to handle noise rates greater than half.

Zhang and Zuo [2008] present an extensive survey on the classification problem with positive and unlabeled feedback. They divide the solution methods into three families described below:

- The first family of methods employs two steps. In the first step, they extract the negative examples from unlabeled data. In the second step, they iteratively apply a classification algorithm to build a set of classifiers and then they select a suitable classifier from that set. Liu et al. [2003], Liu et al. [2002], Yu et al. [2002] and Li and Liu [2003] give examples of the first family of methods.

- The second family of methods estimates statistical queries over positive and unlabeled examples. The methods provided by Denis et al. [2002] and Denis et al. [2003] belong to the second family.

- The third family of methods reduces the problem to learning with high one-sided noise by treating the unlabeled set as noisy negative examples. Some of these methods are given in Lee and Liu [2003b] and Zhang [2005].

Next we study scenarios in which erroneous feedback is provided for a subset of instances i.e. the feedback is noisy. Frénay and Verleysen [2014] note that, in the literature, two types of noise are distinguished: feature noise and label noise.

Feature noise affects the observed values of the feature and label noise affects the observed label of an instance. Firstly we study the previous work done in learning with feature noise.

7.4.2 Learning in the presence of feature noise

Globerson and Roweis [2006] consider the classification problem in which some of the features present during the training phase are potentially deleted by an adversary later. The adversary is permitted to delete only up-to a fixed number features at any given time. The authors formulate this classification problem as a two-player game where the learner attempts to determine the parameters to use for a robust classifier and the adversary tries to delete such features that their deletion causes most harm to the current classifier built by the learner. The authors construct a classifier which is optimal in the worst case deletion scenario.

Dekel et al. [2010] consider a binary classification problem, features of which, available during the training phase, are either deleted or corrupted by an adversary during the classification phase. As we noted in Section 7.2.1, the authors too concede that without limiting the adversary's ability to remove and modify features, any classifier obviously stands no chance of making correct predictions. They overcome this predicament by assigning each feature with an a-priori importance value and assuming that the adversary may remove or corrupt any feature subset whose total value is upper-bounded by a predefined constant. For this setting they devise two approaches given below:

- In the first approach, they formulate the classification problem as a linear program. However the number of constraints in this linear program grows exponentially with the features in the classification problem. They describe a reduction to polynomial-size linear program, which under certain conditions, is an exact equivalent of the original exponential-size linear program. Even when these conditions are not met, the polynomial size linear program is a close approximation of the original linear program. The authors provide an upper bound on the approximation error. They build an efficient classifier

on the polynomial-sized linear program and prove a statistical generalization bound of this approach.

- In the second approach, the authors define an online learning problem in which an adversary removes features from every instance presented to the learner. This online learning problem resembles the original statistical learning problem. A modified version of the Perceptron algorithm (Rosenblatt [1958]) is used to solve the online learning problem. Then this online algorithm is converted into a statistical learning algorithm using an online-to-batch conversion technique.

Having overviewed the relevant work in the topic of feature noise, let us turn our attention to the second type of noise – label noise.

7.4.3 Learning in the presence of label noise

Zhu and Wu [2003] and Sáez et al. [2014] show that label noise is potentially more harmful than feature noise. This is due to the fact that there are many features and the importance of each feature for learning is different whereas there is only one label per instance and it has a large impact on learning. The sources of label noise can be human errors, subjective labeling and problems in data encoding and communication.

In the survey on label noise, Frénay and Verleysen [2014] propose the following taxonomy for label noise:

- Noisy completely at random **(NCAR)**: when the noise is independent of the labels.
- Noisy at random **(NAR)**: when the noise pattern is dependent on the labels.
- Noisy not at random **(NNAR)**: when the feedback corruption is set by an adversary.

Angluin and Laird [1988] consider the classification problem where NCAR errors are introduced in the data during the learning phase. The instances are assumed to

be generated by a sampling procedure that first produces a correctly classified instance; subsequently the instance is subjected to a noise process which introduces random errors before being presented to the learning algorithm. The noise affects each instance independently. The authors introduce a simple model of noise called the *Classification Noise Process* in which each individual instance is reported with incorrect class to the learning algorithm with certain probability. The goal of the learning algorithm is to produce a probably approximately correct classification. To this end, the authors prove that the strategy of selecting the most consistent rule for the input examples is sufficient and usually with feasibly small sample complexity if the errors are only present in less than half the examples on average. The authors also provide a general upper bound on the size of a sample sufficient for learning in finite domains in the presence of classification noise, and evidence that computationally feasible algorithms exist for learning in the presence of classification noise in non-trivial domains.

Natarajan et al. [2013] consider the problem of classification with NAR errors. They assume that the data consists of iid samples from a clean distribution but the feedback given to the learner is drawn from a noisy version of the distribution and the noise rates are dependent on the class labels. The goal of the learner is to minimize a loss function. The authors provide two methods to modify the loss function in such a way that minimizing the sample average of the modified loss function leads to provable risk bounds.

The adversarial NNAR case is particularity noteworthy because positive results in the adversarial model extend to all mistakes and NNAR models the situations in which corruption is due to a deliberate action rather than a random mistake. This model was studied in Kearns and Li [1993]. As we emphasized earlier, the adversary's ability to corrupt the feedback needs to be curbed in order to have learning possible. In this model, the adversary gets to arbitrarily corrupt an instance with a fixed error probability independent of other instances. For this setting, the authors provide bounds on the *optimal malicious error rate* $E_{MAL}(C)$ for the representation

class C i.e. the largest error probability for which any learning algorithm is applicable on C. The upper bound on $E_{MAL}(C)$ conveys a hardness result placing limitations on the rate of error that can be tolerated. The lower bounds on $E_{MAL}(C)$ are given by the way of providing algorithms that tolerate certain rate of error. The authors provide a valuable insight that tolerating errors up-to certain level need not compromise the efficiency of the algorithms. They provide the examples of representation classes for which the optimal error rate can be achieved by polynomial-time algorithms.

More recently, this model was also studied in Feige et al. [2015]. They constrain the adversary so that it can corrupt an input to a certain number of values. In their setting, there is an unknown distribution over the uncorrupted inputs, the learner receives a sample of uncorrupted examples (inputs and labels) and selects a hypothesis (or a mixture of hypotheses) from some limited hypothesis class, mapping a corrupted input to a prediction (or to a distribution over predictions). The error is defined on future inputs which are corrupted by the adversary. They assume the availability of an oracle which on a set of examples finds a minimizing hypothesis in the hypothesis class, i.e. an ERM (Empirical Risk Minimization) oracle. They propose an algorithm based on the idea of adaptive game playing (Freund and Schapire [1999a]), and using a variation of a regret algorithm of Cesa-Bianchi et al. [2007] to find near optimal learner and adversary policies, for the specific sample. Thus, they reduce learning from uncorrupted inputs to learning from corrupted inputs near optimally.

Next we look at the setting where noise is deliberately added to increase data privacy. This motivation of adding noise is related to our setting because corrupt bandits too can be used to achieve a particular kind of privacy as we show in Section 10.3.

7.4.4 Noise addition for data privacy

Noise for the purpose of data privacy could be additive as considered by Kim [1986] in which random stochastic noise is added to the confidential attributes of the data to conceal the distinguishing values. Ciriani et al. [2007] consider the scenario of

adding uncorrelated additive noise which preserves the mean and covariance of the original data but not the correlation coefficients and variances. In the same article, correlated additive noise that preservers the mean and the correlation coefficients of the original data is considered too.

Noise could also be multiplicative as outlined by Kim et al. [2003]. They generate random numbers with mean $= 1$ which are then multiplied to the data instances. They also make use of logarithmic multiplicative noise by adding the random numbers to the logarithm of the original data instances. Mivule [2013] note the trade-off between utility and privacy; closer the perturbed data is to the original, the less confidential that data set becomes, and more distant the perturbed data set is from the original, the more secure but then, utility of the data set might be lost when the statistical characteristics of the origin data set are lost. In section 10.3, when we describe how corrupt bandits can be used to provide privacy, we shall revisit this point.

Presently, we conclude the review of the related work and move on to the next chapter where we provide the lower bounds.

Chapter 8

The Lower Bounds

In this chapter, we shall see the lower bound on the performance measures for best arm identification as well as exploration-exploitation setting. In , we provide first the lower bound on the sample complexity for best arm identification. Then, we provide the lower bound on the cumulative regret for exploration-exploitation setting in Section 8.2.

8.1 Lower bound on the sample complexity for best arm identification

As mentioned in section 1.2.5, the goal of an algorithm in the best arm identification setting is to output the best arm for the given bandit model in the shortest possible time. The performance measure of the algorithm in the fixed confidence setting is the sample complexity τ i.e. the time it takes to identify and output the best arm within the given error probability δ.

Definition 8.1. *An algorithm, which given any valid input, produces the correct output with the probability of at least* $1 - \delta$ *is called a* δ*-correct algorithm.*

We proceed to provide the lower bound on the sample complexity of a δ-correct algorithm for the task of best arm identification for corrupt bandits.

Theorem 8.1. *For a* K*-armed Bernoulli corrupt bandit problem using corruption functions* $g_1, \ldots, g_K$ *with Lipschitz constant* $1/\sigma$*, the expected sample complexity of any* $\delta - correct$

algorithm is lower-bounded as,

$$\mathbb{E}(\tau) \geq \frac{\sigma^2 \cdot d(\delta, 1-\delta)}{2} \sum_{a=1}^{K} \frac{1}{\Delta_a^2}$$

where μ_a is the mean reward for arm a, μ_* is the optimal mean reward, $\Delta_a := \mu_* - \mu_a$ and $d(x,y) := x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the binary relative entropy, with the convention that $d(0,0) := d(1,1) := 0$

Proof. let us define a few notations first. Let $\boldsymbol{\nu}$ characterize a bandit model with reward distribution ν_a for an arm a. Let $\mu_a^{\boldsymbol{\nu}}$ be the mean reward of arm a under bandit model $\boldsymbol{\nu}$ and $\lambda_a^{\boldsymbol{\nu}}$ be the mean feedback of arm a under the same model. Let $a_*(\boldsymbol{\nu}) \in \arg\max \mu_a^{\boldsymbol{\nu}}$ be the optimal arm of bandit model $\boldsymbol{\nu}$.

To obtain a lower bound we adapt a *change-of-distribution* argument from Kaufmann et al. [2016]. We hence consider two K-armed corrupted bandit models, respectively $\boldsymbol{\nu}$ and $\boldsymbol{\nu}'$, with different optimal arms i.e. such that $a_*(\boldsymbol{\nu}) \neq a_*(\boldsymbol{\nu}')$. Let ALG be a δ-correct algorithm for the best arm identification for corrupt bandits with sample complexity τ. Let $\hat{a}_\tau$ represent the arm returned by the algorithm ALG. Let $N_a(t)$ be the number of times arm a has been pulled till time t.

The following lemma can be extracted from Garivier et al. [2016]

Lemma 8.1. *Let $\boldsymbol{\nu}$ and $\boldsymbol{\nu}'$ be two bandit models with K arms and and $T \in \{0\} \cup \mathbb{N}$, then:*

$$\sum_{a=1}^{K} \mathbb{E}_{\boldsymbol{\nu}}[N_a(T)] \cdot KL(\lambda_a^{\boldsymbol{\nu}}, \lambda_a^{\boldsymbol{\nu}'}) \geq d\left(\mathbb{E}_{\boldsymbol{\nu}}(Z), \mathbb{E}_{\boldsymbol{\nu}'}(Z)\right)$$

where $d(x,y) := x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the binary relative entropy and $Z \in [0,1]$ is a random variable measurable from the past-observations filtration $(\mathcal{F}_T)$

Let $Z_a := \mathbb{1}_{\hat{a}_\tau = a}$ be the binary random variable for the event *"ALG's output is a"*. Using lemma 8.1 on Z, we obtain:

$$\begin{aligned}
&\sum_{a=1}^{K} \mathbb{E}_{\boldsymbol{\nu}}[N_a(\tau)] \cdot KL(\lambda_a^{\boldsymbol{\nu}}, \lambda_a^{\boldsymbol{\nu}'}) \\
&\geq d\left(\mathbb{E}_{\boldsymbol{\nu}}[Z_{a_*(\boldsymbol{\nu})}], \mathbb{E}_{\boldsymbol{\nu}'}[Z_{a_*(\boldsymbol{\nu})}]\right) \\
&= d\left(\mathbb{P}_{\boldsymbol{\nu}}[\hat{a}_\tau = a_*(\boldsymbol{\nu})], \mathbb{P}_{\boldsymbol{\nu}'}[\hat{a}_\tau = a_*(\boldsymbol{\nu})]\right)
\end{aligned}$$

Since algorithm A is δ-correct and $a_*(\boldsymbol{\nu}) \neq a_*(\boldsymbol{\nu}')$

$$\sum_{a=1}^{K} \mathbb{E}_{\boldsymbol{\nu}}[N_a(\tau)] \cdot KL(\lambda_a^{\boldsymbol{\nu}}, \lambda_a^{\boldsymbol{\nu}'}) \geq d(\delta, 1-\delta) \tag{8.1}$$

let us assume for the sake of readability that arm 1 is the optimal arm in bandit model $\boldsymbol{\nu}$ (i.e. $a_*(\boldsymbol{\nu}) = 1$). To compute $N_a(\tau)$ for each a such that $2 \leq a \leq K$, we can have $\boldsymbol{\nu}'$ for some $\omega' > 0$ such that

$$\mu_b^{\boldsymbol{\nu}'} = \begin{cases} \mu_1^{\boldsymbol{\nu}} + \omega', & \text{if } b = a \\ \mu_b^{\boldsymbol{\nu}} & \text{otherwise} \end{cases}$$

This translates to the following change in feedback,

$$\lambda_b^{\boldsymbol{\nu}'} = \begin{cases} g_b(\mu_1^{\boldsymbol{\nu}} + \omega') = \lambda_1^{\boldsymbol{\nu}} + \omega, & \text{if } b = a \\ g_b(\mu_b^{\boldsymbol{\nu}}) = \lambda_b^{\boldsymbol{\nu}} & \text{otherwise} \end{cases}$$

where $\omega > 0$ if g_b is increasing and $\omega < 0$ otherwise.

Therefore, the divergence being null for any $b \neq a$, from (8.1) we get:

$$\mathbb{E}_{\boldsymbol{\nu}}[N_a(\tau)] \geq \frac{d(\delta, 1-\delta)}{KL(\lambda_a^{\boldsymbol{\nu}}, \lambda_1^{\boldsymbol{\nu}} + \omega)} \tag{8.2}$$

To compute $N_1(\tau)$, we can have $\boldsymbol{\nu}'$ for some $\omega' > 0$ such that

$$\mu_b^{\boldsymbol{\nu}'} = \begin{cases} \mu_2^{\boldsymbol{\nu}} - \omega', & \text{if } b = 1 \\ \mu_b^{\boldsymbol{\nu}} & \text{otherwise} \end{cases}$$

This translates to the following changes in feedback,

$$\lambda_b^{\boldsymbol{\nu}'} = \begin{cases} g_b(\mu_2^{\boldsymbol{\nu}} - \omega') = \lambda_2^{\boldsymbol{\nu}} - \omega, & \text{if } b = 1 \\ g_b(\mu_b^{\boldsymbol{\nu}}) = \lambda_b^{v} & \text{otherwise} \end{cases}$$

Therefore,

$$\mathbb{E}_{v}[N_1(\tau)] \geq \frac{d(\delta, 1-\delta)}{KL(\lambda_1^{\boldsymbol{\nu}}, \lambda_2^{\boldsymbol{\nu}} - \omega)} \tag{8.3}$$

let us compute the expected value of stopping time τ

$$\begin{aligned}\mathbb{E}(\tau) &= \sum_{a=1}^{K} \mathbb{E}(N_a(\tau)) \\ &\geq \Big[\frac{1}{KL(\lambda_1^{\boldsymbol{\nu}}, \lambda_2^{\boldsymbol{\nu}})} + \sum_{a=2}^{K} \frac{1}{KL(\lambda_1^{\boldsymbol{\nu}}, \lambda_a^{\boldsymbol{\nu}})}\Big] \cdot d(\delta, 1-\delta) \\ &\geq \Big[\frac{1}{2(\lambda_1^{\boldsymbol{\nu}} - \lambda_2^{\boldsymbol{\nu}})^2} + \sum_{a=2}^{K} \frac{1}{2(\lambda_1^{\boldsymbol{\nu}} - \lambda_a^{\boldsymbol{\nu}})^2}\Big] \cdot d(\delta, 1-\delta)\end{aligned}$$

If g_a is $(1/\sigma)$-Lipschitz, we can relate the *reward gap* $\Delta_a^{\boldsymbol{\nu}} = \mu_1^{\boldsymbol{\nu}} - \mu_a^{\boldsymbol{\nu}}$ to the *feedback gap* $\lambda_1^{\boldsymbol{\nu}} - \lambda_a^{\boldsymbol{\nu}}$ by:

$$|\lambda_1^{\boldsymbol{\nu}} - \lambda_a^{\boldsymbol{\nu}}| \leq \Delta_a^{\boldsymbol{\nu}}/\sigma$$

Hence, dropping the $\boldsymbol{\nu}$ superscripts, and with the convention that $\Delta_1 := \Delta_2$, we have:

$$\mathbb{E}(\tau) \geq \frac{\sigma^2 \cdot d(\delta, 1-\delta)}{2} \sum_{a=1}^{K} \frac{1}{\Delta_a^2}$$

□

8.2 Lower bound on the cumulative regret for exploration-exploitation setting

The goal of an algorithm in the exploration-exploitation setting is to minimise the cumulative regret as explianed in Section 1.2.4. Naturally, the performance measure of the algorithm is the expected cumulative regret $\mathrm{CRegret}$. Following a definition by Lai and Robbins [1985] for the classical MAB problem, we define an *uniformly efficient* algorithm for the corrupt bandit problem with prescribed corruption functions $\{g_a\}$ as an algorithm which, for any problem instance ν, $\mathrm{CRegret}_T(\nu) = o(T^\alpha)$ for all $\alpha \in]0, 1[$. Theorem 8.2 provides a lower bound on the regret of an uniformly efficient algorithm.

Theorem 8.2. *Given strictly monotonic corruption functions $\{g_a\}_{a\in A}$, any uniformly efficient algorithm for a Bernoulli corrupt bandit problem satisfies at time horizon T,*

$$\liminf_{T\to\infty} \frac{\mathrm{CRegret}_T}{\log(T)} \geq \sum_{a=2}^{K} \frac{\Delta_a}{d\left(\lambda_a, g_a(\mu_1)\right)}$$

where $d(x,y) :=$ Kullback-Leibler divergence of $(Bernoulli(x), Bernoulli(y))$

Proof. To obtain a lower bound on the regret, we use a *change-of-distribution* argument. Let $\boldsymbol{\nu}$ and $\boldsymbol{\nu'}$ be K-armed corrupted bandit models with different optimal arms i.e. $a_*(\boldsymbol{\nu}) \neq a_*(\boldsymbol{\nu})$. For the ease of readability, let us assume without loss of generality that $a_*(\boldsymbol{\nu}) = 1$.

The log-likelihood ratio of the observations up to time T under $\boldsymbol{\nu}$ and $\boldsymbol{\nu}$, $L_T(\boldsymbol{\nu}, \boldsymbol{\nu'})$, can be written

$$L_T(\boldsymbol{\nu}, \boldsymbol{\nu'}) = \sum_{a=1}^{K} \sum_{s=1}^{N_a(T)} \log \frac{f_{\lambda_a^{\boldsymbol{\nu}}(F_{a,s})}}{f_{\lambda_a^{\boldsymbol{\nu'}}}(F_{a,s})},$$

where $f_x(\cdot)$ denotes the Bernoulli density of mean x and $F_{a,s}$ are the successive observed feedback from arm a. By Wald's lemma on the cumulative sum of random variables (Wald [1944]),

$$\mathbb{E}_{\boldsymbol{\nu}}\left[L_T(\boldsymbol{\nu}, \boldsymbol{\nu'})\right] = \sum_{a=1}^{K} \mathbb{E}_{\boldsymbol{\nu}}[N_a(T)] d(\lambda_a^{\boldsymbol{\nu}}, \lambda_a^{\boldsymbol{\nu'}}),$$

Using Lemma 8.1 with $Z = \frac{N_1(T)}{T}$, one obtains

$$\begin{aligned} &\sum_{a=1}^{K} \mathbb{E}_{\nu}(N_a(T)) d\left(\lambda_a^{\nu}, \lambda_a^{\nu'}\right) \\ &\geq d\Big(\frac{\mathbb{E}_{\nu}(N_1(T))}{T}, \frac{\mathbb{E}_{\nu'}(N_1(T))}{T}\Big) \end{aligned} \tag{8.4}$$

Using the inequality $d(p,q) \geq p\log(1/q) - \log(2)$ (see Garivier et al. [2016]) yields

$$\begin{aligned} &d\Big(\frac{\mathbb{E}_{\boldsymbol{\nu}}(N_1(T))}{T}, \frac{\mathbb{E}_{\boldsymbol{\nu'}}(N_1(T))}{T}\Big) \\ &\geq \frac{\mathbb{E}_{\boldsymbol{\nu}}(N_1(T))}{T} \log\left(\frac{T}{\mathbb{E}_{\boldsymbol{\nu'}}(N_1(T))}\right) - \log(2) \end{aligned}$$

Since $a_*(\boldsymbol{\nu}) = 1$, and $a_*(\boldsymbol{\nu'}) \neq 1$, $\mathbb{E}_{\boldsymbol{\nu}(N_1(T))\sim T}$ and $\mathbb{E}_{\boldsymbol{\nu'}}(N_1(T)) = o(T^\alpha)$ for all $\alpha \in]0, 1]$. Hence one can show that

$$\frac{\mathbb{E}_{\boldsymbol{\nu}(N_1(T))}}{T} \sim 1 \text{ and } \log\left(\frac{T}{\mathbb{E}_{\nu'}[N_1(T)]}\right) \sim \log(T).$$

Equation (8.4) yields

$$\liminf_{T\to\infty} \frac{\sum_{a=1}^K \mathbb{E}_{\boldsymbol{\nu}}(N_a(T)) d\left(\lambda_a^{\boldsymbol{\nu}}, \lambda_a^{\boldsymbol{\nu'}}\right)}{\log T} \geq 1. \tag{8.5}$$

To obtain a lower bound on $\mathbb{E}_{\boldsymbol{\nu}}[N_a(T)]$ for each $a \in \{2, \ldots, K\}$, one can choose $\boldsymbol{\nu'}$ such that, for some $\epsilon > 0$,

$$\mu_b^{\boldsymbol{\nu'}} = \begin{cases} \mu_1^{\boldsymbol{\nu}} + \epsilon, & \text{if } b = a \\ \mu_b^{\boldsymbol{\nu}} & \text{otherwise} \end{cases}$$

This translates to the following change in feedback,

$$\lambda_b^{\boldsymbol{\nu'}} = \begin{cases} g_b(\mu_1^{\boldsymbol{\nu}} + \epsilon) & \text{if } b = a, \\ g_b(\mu_b^{\boldsymbol{\nu}}) = \lambda_b^{\boldsymbol{\nu}} & \text{otherwise.} \end{cases}$$

As $d\left(\lambda_b^{\boldsymbol{\nu}}, \lambda_b^{\boldsymbol{\nu'}}\right) = 0$ for $b \neq a$, using equation (8.5) we get

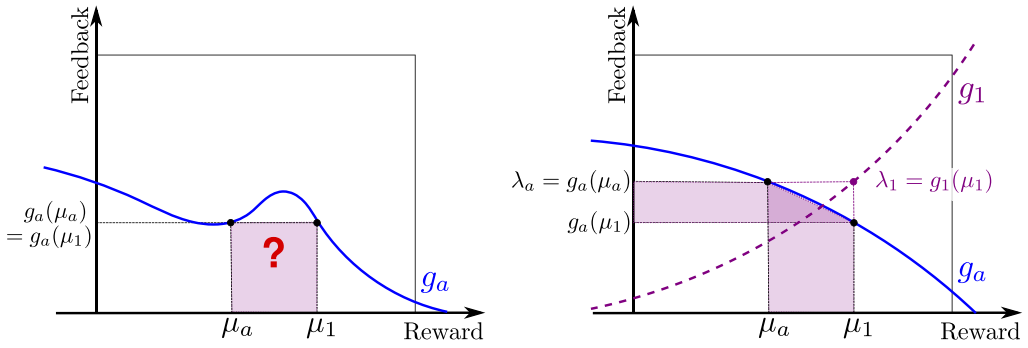
$$\liminf_{T\to\infty} \frac{\mathbb{E}_{\boldsymbol{\nu}}(N_a(T))}{\log T} \geq \frac{1}{d\left(\lambda_a^{\boldsymbol{\nu}}, g_a(\mu_1 + \epsilon)\right)}$$

Letting ϵ go to zero for each $a \in \{2, \ldots, K\}$ (and using that $\{g\}_{a=1}^K$ are continuous), one obtains,

$$\liminf_{T\to\infty} \frac{\mathrm{CRegret}_T(\boldsymbol{\nu})}{\log(T)} \geq \sum_{a=2}^K \frac{\Delta_a^{\boldsymbol{\nu}}}{d\left(\lambda_a^{\boldsymbol{\nu}}, g_a(\mu_1^{\boldsymbol{\nu}})\right)}.$$

□

The lower bound reveals that the divergence between the mean feedback from $a \in A$ and the image of the optimal reward μ_1 with g_a plays a crucial role in distinguishing arm a from the optimal arm. The shape of the g_a function in the neighborhood of both μ_a and μ_1 has a great impact on the information the learner can



(A) Uninformative g_a function. (B) Informative g_a function.

FIGURE 8.1: In Figure 8.1a, g_a is such that $\lambda_a = g_a(\mu_1)$ thereby making it impossible to discern arm a from the optimal arm given the mean feedback. In Figure 8.1b, a steep monotonic g_a leads the reward gap $\Delta_a = \mu_1 - \mu_a$ into a clear gap between λ_a and $g_a(\mu_1)$.

extract from the received feedback. Particularly, if the g_a function is non-monotonic, as shown in Figure 8.1a, it might be impossible to distinguish between arm a and the optimal arm. To dodge this problem, we assume the corruption functions $\{g_a\}_{a \in A}$ to be strictly monotonic in our algorithms and we denote its corresponding inverse function by g_a^{-1}. Such an informative corruption function is shown in Figure 8.1b. To clarify that the gap between λ_a and λ_1 is not relevant here, we also plot in Figure 8.1b a corruption function g_1 which differs from g_a and causes fortuitously the two arms to have the same mean feedback with different interpretations in terms of mean rewards.

Having proven the lower bounds on the performance measures of best arm identification and exploration-exploitation, we shall see the algorithms for these two settings.

Chapter 9

The Algorithms and their Analyses

In this chapter, we shall present the algorithms for various settings of corrupt bandits. Let us begin with the setting of best arm identification. Firstly, in Section 9.1, we introduce two algorithms for the corrupt bandits for best arm identification with fixed confidence. In Section 9.2, we introduce two algorithms for the corrupt bandits in the exploration-exploitation setting.

9.1 Algorithms for best arm identification

As mentioned in Section 1.4.1, best arm identification algorithms for the classical bandit problem usually proceed in rounds eliminating sub-optimal arms. Using similar round-based format, we present two elimination algorithms for best arm identification in corrupt bandits.

9.1.1 Median elimination for corrupt bandits

The first algorithm we present is derived from the corresponding median elimination algorithm for classical bandits presented in Even-Dar et al. [2006]. The goal of ME-CF(ϵ, δ) is to output an ϵ-approximate best arm with probability at-least $1-\delta$. An ϵ-approximate best arm is any arm whose mean reward is at most ϵ lower than that of the best arm.

In each round, ME-CF (21) samples every remaining arm equal number of times which is determined using the parameters ϵ and δ. Then, it computes the empirical estimates for each remaining arm applying the corresponding inverse corruption function on the mean empirical feedback. Only the arms whose empirical estimate is

Algorithm 21 Median elimination for corrupt bandits (ME-CF)

Input: A bandit model with a set of arms $A \coloneqq \{1, \ldots, K\}$ arms with unknown reward means $\mu_1, \ldots, \mu_K$ and unknown feedback means $\lambda_1, \ldots, \lambda_K$ and, strictly monotonic and differentiable corruption functions $g_1, \ldots, g_k$ with Lipschtiz constant $\frac{1}{\sigma}$.
Parameters: $\epsilon > 0$, $\delta > 0$
Set $S_1 \leftarrow A, \epsilon_1 \leftarrow \epsilon/4, \delta \leftarrow \delta/2, l \leftarrow 1$
Do
    Sample every arm a in S_l for $1/(\epsilon_l/2\sigma)^2 \log(3/\delta_l)$ times and let $\hat{\lambda}_a^l$ be its mean empirical feedback and let $\hat{f}_a^l \coloneqq g_a^{-1}(\hat{\lambda}_a^l)$ represent its empirical estimate.
    Let Med_l be the median of $\{\hat{f}_1^l\}_{a \in S_l}$.
    $S_{l+1} \leftarrow S_l \setminus \{a : \hat{f}_a^l < \text{Med}_l\}$
    $\epsilon_{l+1} \leftarrow \frac{3}{4}\epsilon_l, \delta_{l+1} \leftarrow \frac{\delta}{2}, l \leftarrow l + 1$
Until $|S_l| = 1$
Output the arm in S_l

greater than the median of the empirical estimates of the remaining arms are carried forward to the next round. This process repeats till only a single arm remains at which point the algorithm stops and returns the sole remaining arm. The following theorem bounds the sample complexity of ME-CF.

Theorem 9.1. *For a K-armed corrupt bandit problem using strictly monotonic and differentiable corruption functions $g_1, \ldots, g_K$ with Lipschitz constant $1/\sigma$,* ME-CF *returns an ϵ-approximate arm with probability at-least $1-\delta$ and its sample complexity is upper-bounded by*

$$\mathbb{E}(\tau) \leq O\left(\frac{K\sigma^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right)$$

This proof proceeds along the lines of the proof for the sample complexity for the median elimination algorithm given in Even-Dar et al. [2006] and it is given in Appendix B.

9.1.2 Exponential-gap elimination for corrupt bandits

The next algorithm we present is derived from the exponential-gap elimination algorithm given in Karnin et al. [2013]. The goal of EGE-CF (22) is to output the best arm with probability at-least $1 - \delta$ i.e. it is an δ-correct algorithm.

During each round, EGE-CF too samples every remaining arm an equal number of times which is determined using the parameter δ. Then, it computes the empirical estimates for each arm applying the corresponding inverse corruption function

Algorithm 22 Exponential-gap elimination for corrupt bandits (EGE-CF)

Input: A bandit model with a set of arms $A := \{1, \ldots, K\}$ arms with unknown reward means $\mu_1, \ldots, \mu_K$ and unknown feedback means $\lambda_1, \ldots, \lambda_K$ and, strictly monotonic and differentiable corruption functions $g_1, \ldots, g_k$ with Lipschtiz constant $\frac{1}{\sigma}$.
Parameters: δ
Set $S_1 \leftarrow A, l \leftarrow 1$
While $|S_l| > 1$
    let $\epsilon_l \leftarrow 2^{-l}/4$ and $\delta_l \leftarrow \delta/(50l^3)$
    Sample each arm $a \in S_l$ for $2/(\epsilon_l/\sigma)^2 \log(1/\delta_l)$ and let $\hat{\lambda}_a^l$ be its mean empirical feedback and let $\hat{f}_a^l := g_a^{-1}(\hat{\lambda}_a^l)$ represent its empirical estimate.
    Invoke $a_l \leftarrow$ ME-CF$(S_l, \epsilon_l/2, \delta_l)$
    Set $S_{l+1} \leftarrow S_l \setminus \{a \in S_l : \hat{f}_a^l < \hat{f}_{a_l}^l - \epsilon_l\}$
    $l \leftarrow l + 1$
End while
Output the arm in S_l

on the mean empirical feedback. It uses ME-CF as a subroutine to select an approximately optimal arm from the set of remaining arms. Only the arms whose empirical estimates are sufficiently close to the approximately correct arm are carried forward to the next round. Arms are eliminated in successive rounds till only a single arm remains at which point the algorithm stops and the remaining arm is returned as the output. The following theorem bounds the sample complexity of EGE-CF.

Theorem 9.2. *For a K-armed corrupt bandit problem using strictly monotonic and differentiable corruption functions $g_1, \ldots, g_K$ with Lipschitz constant $1/\sigma$,* EGE-CF *returns the optimal arm with probability at-least $1 - \delta$ and its expected sample complexity is upper-bounded by*

$$\mathbb{E}\tau \leq O\left(\sigma^2 \sum_{a=2}^{K} \frac{1}{\Delta_a^2} \log\left(\frac{1}{\delta} \log \frac{1}{\Delta_a}\right)\right)$$

The proof for Theorem 9.2 is given in appendix C

9.2 Algorithms for exploration-exploitation setting

As explianed in Section 1.2.4, the goal of an algorithm in this setting is to minimise the cumulative regret. There are two popular approaches to solve the MAB problem in this setting: the frequentist approach and the Bayesian approach. Some of the notable frequentist algorithms are UCB1(Auer et al. [2002a]), KL-UCB (Cappé et al.

[2013]), while one of the earliest solutions for the MAB problem known as THOMPSON SAMPLING (Thompson [1933]) is a Bayesian algorithm. We too shall solve the corrupt bandits problem with both a frequentist and a Bayesian approach.

9.2.1 kl-UCB for MAB with corrupted feedback (kl-UCB-CF)

We propose an algorithm called kl-UCB-CF which is an adaptation of the KL-UCB algorithm of Cappé et al. [2013]. It computes $\mathrm{Index}_a(t)$ from a KL-based confidence interval on λ_a (line 3 in algorithm 23) to be used as an upper-confidence bound on μ_a. We denote by $\hat{\lambda}_a(t)$ the empirical mean of the feedback obtained from the arm a until time t.

Algorithm 23 kl-UCB for MAB with corrupted feedback (kl-UCB-CF)

Input: A bandit model with a set of arms $A \coloneqq \{1, \dots, K\}$ arms with unknown mean rewards $\mu_1, \dots, \mu_K$ and unknown mean feedbacks $\lambda_1, \dots, \lambda_K$ and monotonic and continuous corruption functions $g_1, \dots, g_k$.
Parameters: A non-decreasing (exploration) function $f : \mathbb{N} \to \mathbb{R}$, $d(x, y) \coloneqq \mathrm{KL}(\mathcal{B}(x), \mathcal{B}(y))$, Time horizon T.

1: **Initialization:** Pull each arm once.
2: **for** time $t = K, \dots, T-1$ **do**
3: Compute for each arm a in A the quantity

$$\mathrm{Index}_a(t) \coloneqq \max\left\{q : \ N_a(t) \cdot d(\hat{\lambda}_a(t), g_a(q)) \leq f(t)\right\}$$

4: Pull arm $\hat{a}_{t+1} \coloneqq \underset{a}{\mathrm{argmax}}\ \mathrm{Index}_a(t)$ and observe the feedback F_{t+1}.
5: **end for**

Theorem 9.3 gives an upper bound on the regret of kl-UCB-CF showing that it matches the lower bound given in Theorem 8.2. A more explicit finite-time bound is proved in Appendix D.

Theorem 9.3. kl-UCB-CF *using* $f(t) \coloneqq \log(t) + 3\log(\log(t))$ *on a* K*-armed Bernoulli corrupt bandit with strictly monotonic and continuous corruption functions* $\{g_a\}_{a \in A}$ *satisfies at time* T*,*

$$\mathrm{CRegret}_T \leq \sum_{a=2}^{K} \frac{\Delta_a \log(T)}{d\left(\lambda_a, g_a(\mu_1)\right)} + O(\sqrt{\log(T)}).$$

Here we present a brief sketch of the proof. The detailed version of this proof is given in Appendix D.

Proof sketch: Recall that $N_a(t) :=$ the number of times arm a has been pulled till time t. Letting $F_{a,s}$ being the successive observations of arm a and $\hat{\lambda}_{a,s} := \frac{1}{s}\sum_{l=1}^{s} F_{a,l}$, one has $\hat{\lambda}_a(t) = \hat{\lambda}_{k,N_a(t)}$ when $N_a(t) > 0$.

The event $\{\hat{a}_{t+1} = a\}$ can happen either with λ_1 being within or outside its KL-based confidence interval. We compute the probability of λ_1 being outsides its interval i.e. either greater than $u_1(t)$ or lower than $l_1(t)$ depending upon whether g_1 is increasing or decreasing respectively. If λ_1 is within its confidence interval, the fact that arm a is played translates into the upper bound (resp. lower bound) on λ_a being greater (resp. lower) than $g_a(\mu_1)$ when g_a is increasing (resp. decreasing). We compute all the required quantities to get an upper bound on $\mathbb{E}[N_a(T)]$ which when multiplied by Δ_a and summed over all the non-optimal arms gives an upper bound on the expected regret. Starting from

$$\mathbb{E}(N_a(T)) = 1 + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a),$$

we provide below the decomposition that are used in each case.

- When both g_1 and g_a are increasing,

$$\mathbb{E}(N_a(T)) \leq 1 + \sum_{t=K}^{T-1} \mathbb{P}(u_1(t) < \lambda_1) + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1))$$

- When g_1 is decreasing and g_a is increasing,

$$\mathbb{E}(N_a(T)) \leq 1 + \sum_{t=K}^{T-1} \mathbb{P}(\ell_1(t) > \lambda_1) + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1))$$

- g_1 is increasing and g_a is decreasing,

$$\mathbb{E}(N_a(T)) \leq 1 + \sum_{t=K}^{T-1} \mathbb{P}(u_1(t) < \lambda_1) + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \ell_a(t) \leq g_a(\mu_1))$$

- When g_1 is decreasing and g_a is decreasing,

$$\mathbb{E}(N_a(T)) \leq 1 + \sum_{t=K}^{T-1} \mathbb{P}(\ell_1(t) > \lambda_1) + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \ell_a(t) \leq g_a(\mu_1))$$

Using deviation inequalities (introduced by Cappé et al. [2013]), we show that

$$\sum_{t=K}^{T-1} \mathbb{P}(u_1(t) < g_1(\mu_1)) \leq 3 + 4e \log(\log T) \in o(\log T) \tag{9.1}$$

$$\sum_{t=K}^{T-1} \mathbb{P}(\ell_1(t) > g_1(\mu_1)) \leq 3 + 4e \log(\log T) \in o(\log T) \tag{9.2}$$

We introduce the notation $d^+(x,y) \coloneqq d(x,y)\mathbb{1}_{(x<y)}$ and $d^-(x,y) \coloneqq d(x,y)\mathbb{1}_{(x>y)}$, where $\mathbb{1}$ is the indicator function. So we can write, when g_a is increasing,

$$\sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1)) \leq \sum_{s=1}^{T-1} \mathbb{P}\left(s \cdot d^+(\hat{\lambda}_{a,s}, g_a(\mu_1)) \leq f(T)\right). \tag{9.3}$$

And when g_a is decreasing,

$$\sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \ell_a(t) \leq g_a(\mu_1)) \leq \sum_{s=1}^{T-1} \mathbb{P}\left(s \cdot d^-(\hat{\lambda}_{a,s}, g_a(\mu_1)) \leq f(T)\right). \tag{9.4}$$

The quantity in the right-hand side of (9.3) is upper bounded in Appendix A.2. of Cappé et al. [2013] by

$$\frac{f(T)}{d(\lambda_a, g_a(\mu_1)} + \sqrt{2\pi}\sqrt{\frac{(d'(\lambda_a, g_a(\mu_1)))^2}{(d(\lambda_a, g_a(\mu_1)))^3}}\sqrt{f(T)} + 2\left(\frac{d'(\lambda_a, g_a(\mu_1))}{d(\lambda_a, g_a(\mu_1))}\right)^2 + 1. \tag{9.5}$$

where $d'(x,y)$ is used to indicate the derivative of $d(x,y)$ with respect to the first variable. Following the same approach as Cappé et al. [2013], we can prove that the right-hand side of (9.4) is upper bounded by the same quantity. Combining inequalities (9.1), (9.2), (9.3), (9.4) and (9.5) with the initial decomposition of $\mathbb{E}[N_a(T)]$,

$$\mathbb{E}[N_a(T)] \leq \frac{\log(T)}{d(\lambda_a, g_a(\mu_1))} + \sqrt{2\pi}\sqrt{\frac{d'(\lambda_a, g_a(\mu_1))^2}{d(\lambda_a, g_a(\mu_1))^3}}\sqrt{\log(T) + 3\log\log(T)}$$
$$+ \left(4e + \frac{3}{d(\lambda_a, g_a(\mu_1)}\right)\log\log(T) + 2\left(\frac{d'(\lambda_a, g_a(\mu_1)}{d(\lambda_a, g_a(\mu_1)}\right)^2 + 4.$$

So the expected regret of kl-UCB-CF is at most

$$\sum_{a=2}^{K} \Delta_a \Big[\frac{\log(T)}{D_a} + \sqrt{2\pi}\sqrt{\frac{(D'_a)^2}{D_a^3}}\sqrt{\log(T) + 3\log\log(T)}$$
$$+ \left(4e + \frac{3}{D_a}\right)\log\log(T) + 2\left(\frac{D'_a}{D_a}\right)^2 + 4\Big]$$

where $D_a \coloneqq d(\lambda_a, g_a(\mu_1))$ and $D'_a \coloneqq d'(\lambda_a, g_a(\mu_1))$, which concludes the proof.

□

9.2.2 UCB1 for MAB with corrupted feedback (UCB-CF)

UCB1 (Auer et al. [2002a]) can also be adapted to corrupted feedback by modifying the index to

$$\text{Index}_a(t) \coloneqq \begin{cases} g_a^{-1}\left(\hat{\lambda}_a(t) + \sqrt{\frac{f(t)}{2N_a(t)}}\right) & \text{if increasing } g_a \\ g_a^{-1}\left(\hat{\lambda}_a(t) - \sqrt{\frac{f(t)}{2N_a(t)}}\right) & \text{if decreasing } g_a \end{cases}$$

Corollary 9.1. *With $f(t) \coloneqq \log(t) + 3\log(\log(t))$, the expected cumulative regret of* UCB-CF *on a K-armed corrupt bandit with strictly monotonic and continuous corruption functions $\{g_a\}_{a=1}^K$ at time horizon T is in $O\left(\sum_{a=2}^K \frac{\Delta_a \log(T)}{(\lambda_a - g_a(\mu_1))^2}\right)$.*

The proof of this corollary follows the proof of Theorem 9.3 using the quadratic divergence $2(x-y)^2$ in place of $d(x, y)$. The UCB-CF algorithm is only order optimal with respect to the bound of Theorem 8.2, but its index is simpler to compute.

9.2.3 Thompson sampling for MAB with corrupted feedback (TS-CF)

TS-CF maintains a Beta posterior distribution on the mean feedback of each arm. At round $t+1$, for each arm a, it draws a sample $\theta_a(t)$ from the posterior distribution on λ_a and pulls the arm for which $g_a^{-1}(\theta_a(t))$ is largest. This mechanism ensures that at each round, the probability that arm a is played is the posterior probability of this arm to be optimal, as in regular Thompson Sampling (TS) (Thompson [1933]).

Theorem 9.4. *When* TS-CF *is run on any K-armed corrupt bandit with corruption functions $\{g_a\}_{a=1}^K$, there exists a constant $C_\epsilon \coloneqq C(\epsilon, \{\mu_a\}_{a=1,\dots,K}, \{g_a\}_{a=1,\dots,K})$ for all the sub-optimal arms a, and $\epsilon > 0$, such that the expected cumulative regret at time horizon T is*

$$\mathbb{E}[\text{CRegret}_T] \leq (1+\epsilon) \sum_{a=2}^K \frac{\Delta_a \log(T)}{d(\lambda_a, g_a(\mu_1))} + C_\epsilon.$$

Algorithm 24 Thompson sampling for MAB with corrupted feedback (TS-CF)

Input: A bandit model with a set of arms $A \coloneqq \{1, \dots, K\}$ arms with unknown reward means $\mu_1, \dots, \mu_K$ and unknown feedback means $\lambda_1, \dots, \lambda_K$ and monotonic and continuous corruption functions $g_1, \dots, g_K$.
Parameters: Time horizon T.
Initialization: For each arm a in A, set $\mathrm{success}_a = 0$ and $\mathrm{fail}_a = 0$
for $t = 0, \dots, T-1$ **do**.
  For each arm a in A, sample $\theta_a(t)$ from $\mathrm{Beta}(\mathrm{success}_a + 1, \mathrm{fail}_a + 1)$
  Pull arm $\hat{a}_{t+1} \coloneqq \arg\max_a g_a^{-1}(\theta_a(t))$ and observe the feedback F_{t+1}.
  if $F_{t+1} = 1$ **then**
    $\mathrm{success}_{\hat{a}_{t+1}} = \mathrm{success}_{\hat{a}_{t+1}} + 1$
  else
    $\mathrm{fail}_{\hat{a}_{t+1}} = \mathrm{fail}_{\hat{a}_{t+1}} + 1$
  end if
end for

This theorem also yields the asymptotic optimality of TS-CF with respect to the lower bound given in Theorem 8.2. We give a sketch of its proof here. The detailed version of this proof is given in Appendix E.

Proof sketch. Assume 1 to be the optimal arm. For each arm non-optimal arm a, choose two thresholds u_a and w_a such that $\lambda_a < u_a < w_a < g_a(\mu_1)$ if g_a is increasing and $\lambda_a > u_a > w_a > g_a(\mu_1)$ is decreasing.

Define $E_a^{\lambda}(t)$ as the event $\{g_a^{-1}(\hat{\lambda}_a(t)) \leq g_a^{-1}(u_a)\}$ and $E_a^{\theta}(t)$ as the event $\{g_a^{-1}(\theta_a(t)) \leq g_a^{-1}(w_a)\}$.

Define $\mathcal{F}_t$ as the history of arm selections and received feedbacks including time t and recall that TS-CF selects the arm as follows, $\hat{a}_{t+1} = \mathrm{argmax}_a\ \theta_a(t)$, where $\theta_a(t)$ is a sample from the posterior distribution on arm a after t observations.

Define $p_{a,t} \coloneqq \mathbb{P}(g_1^{-1}(\theta_1(t)) > g_a^{-1}(w_a) \mid \mathcal{F}_t)$.

Starting from the following decomposition,

$$\mathbb{E}[N_a(T)] = \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, E_a^{\lambda}(t), E_a^{\theta}(t)) + \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, E_a^{\lambda}(t), \overline{E_a^{\theta}(t)}) + \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \overline{E_a^{\lambda}(t)}),$$

We provide below some lemmas that permit to bound these three terms. These results generalize to the corrupted setting the main steps of the analysis of Thompson

Sampling by Agrawal and Goyal [2013]. In particular Lemma 9.1 relates the probability of drawing a sub-optimal arm a when the two "typical" events $E_a^\theta(t)$ and $E_a^\lambda(t)$ hold to the probability of playing the optimal arm, through a coefficient involving the inverse of $p_{a,t}$. This result is used in the analysis in conjugation with Lemma 9.4, that bounds the growth of the expected value of $p_{a,t}^{-1}$.

Lemma 9.1. $\mathbb{P}(\hat{a}_{t+1} = a, E_a^\theta(t), E_a^\lambda(t) \mid \mathcal{F}_t) \leq \frac{(1-p_{a,t})}{p_{a,t}} \mathbb{P}(\hat{a}_{t+1} = 1, E_a^\theta(t), E_a^\lambda(t) \mid \mathcal{F}_t)$.

Lemma 9.2. *When g_a is increasing (resp. decreasing), for any $u'_a \in (u_a, w_a)$ (resp. (w_a, u_a)), when T is large enough,*

$$\sum_{t=0}^{T-1} \mathbb{P}\Big(\hat{a}_{t+1} = a, \overline{E_a^\theta(t)}, E_a^\lambda(t)\Big) \leq \frac{\log(T)}{d(u'_a, w_a)} + 1.$$

Lemma 9.3. $\sum_{t=0}^{T-1} \mathbb{P}\Big(\hat{a}_{t+1} = a, \overline{E_a^\lambda(t)}\Big) \leq 1 + \frac{1}{d(u_a, \lambda_a)}$.

Lemma 9.4. *Let τ_s be the instant of the s-th play of arm 1. Then there exists a function $f(s) \coloneqq f(s, \lambda_1, g_1(g_a^{-1}(\mu_1)))$ satisfying $\sum_{s=1}^{\infty} f(s) < \infty$ such that for all s,*

$$\mathbb{E}\left[\frac{1}{p_{a,\tau_s+1}}\right] \leq 1 + f(s).$$

Let (τ_s) be the sequence introduced in Lemma 9.4. Using Lemma 9.1-9.4, one can write, for large enough T,

$$\begin{aligned}
\mathbb{E}[N_a(T)] &\leq \sum_{t=0}^{T-1} \mathbb{E}\left[\frac{(1-p_{a,t})}{p_{a,t}} \mathbb{1}_{(\hat{a}_{t+1}=1, E_a^\theta(t), E_a^\lambda(t))}\right] + \frac{\log T}{d(u'_a, w_a)} + 1 + \frac{1}{d(u_a, \lambda_a)} + 1 \\
&\leq \sum_{s=0}^{T-1} \mathbb{E}\Big[\frac{(1-p_{a,\tau_s+1})}{p_{a,\tau_s+1}} \underbrace{\sum_{t=\tau_s}^{\tau_{s+1}-1} \mathbb{1}_{(\hat{a}_{t+1}=1)}}_{=1}\Big] + \frac{\log T}{d(u'_a, w_a)} + \frac{1}{d(u_a, \lambda_a)} + 2 \\
&\leq \frac{\log T}{d(u'_a, w_a)} + \sum_{s=0}^{\infty} f(s) + \frac{1}{d(u_a, \lambda_a)} + 2.
\end{aligned}$$

Fix $\epsilon > 0$. Using the monotonicity properties of the divergence function d, there exists $u_a < u'_a < w_a$ in the increasing case and $u_a > u'_a > w_a$ in the decreasing case such that $d(u'_a, w_a) \geq d(\lambda_a, g_a(\mu_1))/(1+\epsilon)$. For these particular choice, one obtains

$$\mathbb{E}[N_a(T)] \leq (1+\epsilon)\frac{\log(T)}{d(\lambda_a, g_a(\mu_a))} + R(u_a, u'_a, w_a),$$

where $R(u_a, u'_a, w_a)$ is a rest term that depends on $\epsilon, \mu_1, \mu_a, g_1$ and g_a. The result follows using that $\mathbb{E}[\mathrm{CRegret}_T] = \sum_{a=2}^{K} \Delta_a \mathbb{E}[N_a(T)]$. □

9.2.4 kl-UCB-CF and TS-CF for MAB with randomized response

In this subsection, we consider the application of the dedicated corrupt bandit algorithms on randomized response which is a special form of corrupted feedback introduced in the subsection 7.2.3. The following corollary bounds the expected regret of kl-UCB-CF and TS-CF when a applied on a MAB problem with randomized response.

Corollary 9.2. *The expected regret of* kl-UCB-CF *and* TS-CF *for a K-armed MAB problem with randomized response using corruption matrices $\{\mathbb{M}\}_{a\in A}$ at time horizon T is in $O\Big(\sum_{a=2}^{K} \frac{\log(T)}{\Delta_a (p_{00}(a)+p_{11}(a)-1)^2}\Big)$ where*

$$\mathbb{M}_a \coloneqq \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} p_{00}(a) & p_{01}(a) \\ p_{10}(a) & p_{11}(a) \end{bmatrix} \end{array}$$

and

$$\mathbb{M}_a(x, y) \coloneqq \mathbb{P}(\textit{feedback for arm } a = x \mid \textit{reward for arm } a = y)$$

Proof. This corollary follows from Theorem 9.3 and Theorem 9.4 together with Pinsker's inequality. □

In the next chapter, we shall provide a practical application of the corrupted feedback and explain how the algorithms introduced in this chapter can be used for that purpose.

Chapter 10

Corrupt Bandits to enforce Differential Privacy

In this chapter, we shall explain how our setting can be used for the practical application of enforcing differential privacy. In Section 10.1, we provide an introduction to the topic of differential privacy. In Section 10.2, we first provide the motivation for using differential privacy in bandits and then overview the related previous work. Lastly, in Section 10.3, we present how the corrupt bandits can be used for achieving differential privacy.

10.1 Introduction to differential privacy

Differential privacy was introduced by Dwork et al. [2006]. Dwork and Roth [2014] describe differential privacy as a promise, made by a data holder to a data subject: "You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available." A dataset is a collection of records and a statistic is a quantity computed from a record. A querying mechanism or simply a mechanism is an algorithm that takes as an input a dataset and produces output for a query. The goal of a privacy-preserving querying mechanism is to ensure that the information about individual records are not revealed from the output. The other approaches to privacy-preserving mechanism include removal of personally identifiable information from the records (anonymization) or answering only summary queries. However Dwork and Roth [2014] explain the vulnerabilities of these approaches and they illustrate

how these approaches can be defeated. For example, a *linkage attack* can be used to match anonymized records with non-anonymized records in a different dataset to compromise the personal information from the anonymized records. Moreover the answers to the two summary queries - first for all the individuals in the dataset and second for all the individuals in the dataset except X, does reveal the individual information about X even when only summary queries are answered. This is called as *differencing attack*. To address these concerns, differential privacy produces output from which it is possible to learn the properties of the population of the samples present in the dataset as a whole while not divulging the information about the individual samples. Differential privacy ensures that any sequence of outputs is "essentially" equally likely to occur independent of the presence or the absence of an individual record in a dataset. More formally, it is defined as follows:

Definition 10.1. *Any randomized mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if for all the datasets $d_1, d_2 \in Domain(\mathcal{M})$ differing in at most one record and for all $S \in Range(\mathcal{M})$*

$$\mathbb{P}[\mathcal{M}(d_1) \in S] \leq \exp(\epsilon)\mathbb{P}[\mathcal{M}(d_2) \in S] + \delta$$

If $\delta = 0$, then $\mathcal{M}$ is said to be ϵ-differentially private.

By definition, differential privacy is unsusceptible to differencing attack. Moreover, since differential privacy is a property of the data access mechanism, and is unrelated to the presence or absence of auxiliary information, it is also immune to linkage attacks as well. Dwork [2010] enlist the following strengths of differential privacy:

1. It is independent of any additional information, including other databases, available to the adversary.

2. It is achievable using fairly simple and general mechanisms.

3. It frequently allows very accurate analyses.

In the definition 10.1, we saw that randomization is a necessary condition for a mechanism to be differentially private. let us investigate the prerequisite of randomization for differential privacy.

10.1.1 Why randomization?

Consider for example a simple dataset which stores the credit rating for a number of individuals. A querying mechanism accesses this dataset and returns the arithmetic mean of the values. Consider the following two datasets: dataset A which contains the information about user X and dataset B which doesn't. Any deterministic querying process yields two different outputs for these two datasets. Knowing these two values, an adversary realizing that the dataset is one of these two almost identical datasets can learn the value of the credit rating of user X. Therefore randomization is essential for any privacy guarantee.

We have already seen that differential privacy can be preferable to other privacy mechanisms by the virtue of being immune to privacy-compromising attacks that neutralise the latter. A possible counter-argument for differential privacy might be why not simply use secure cryptosystems which are guaranteed to exist under standard computational assumptions. This counter-argument is however fallacious as explained below.

10.1.2 Why not simply use cryptosystems?

A cryptosystem is a pair of algorithms that take plain-text as a key whose privacy is needed to be protected and convert it to cipher-text which appears to be a random gibberish and back. In a cryptosystem, there are three parties: the message sender (who encrypts the plain-text message), the message receiver (who decrypts the cipher-text), and the eavesdropper whose goal is to read the plain-text message without having the authority to do so. A secure cryptosystem ensures that eavesdropper is foiled in their attempt to compromise the privacy of the sent message. However, in a privacy-preserving mechanism, there are only two parties: the curator who runs the mechanism (analogous to the sender) and the receiver who receives the output to the queries. The receiver tries to discover personal information about the queried records. To wit, the legitimate receiver is the same party as the snooping eavesdropper in a privacy-preserving mechanism. Therefore a secure cryptosystem is not a suitable substitute for a privacy preserving mechanism.

Having seen what does differential privacy mean in the general context, let us turn our attention to differential privacy specifically for the bandit problems.

10.2 Differential privacy in bandits

Since its inception, differential privacy has been used in conjunction with many settings (Dwork [2006], Dwork [2009], Dwork [2010]). Particularly, it has also been applied in the milieu of online learning (Jain et al. [2012] ,Thakurta and Smith [2013]), of which bandit learning is a subset. Before studying how differential privacy is adopted to bandits, let us see the motivation for applying differential privacy to bandit applications.

10.2.1 Motivation for differential privacy in bandits

In section 1.3.2, we looked at Internet advertising as one of the main applications of the bandit problem. To personalise the advertisements for the users, advertising systems make use of personal user information. This opens the way for loss of user privacy. Korolova [2010] provide experimental evidence of several approaches to obtain private user information from the advertising system of the world's largest online social network, Facebook. Calandrino et al. [2011] show methods to infer users' private transactions from the information routinely revealed by the users to the advertising system. The provided methods make use of aggregate statistics which contain no personally identifiable information. The authors have employed these methods on public data extracted from the advertising systems of the popular websites Hunch, Last.fm, LibraryThing, and Amazon. To deal with such type of methods, one of the proposals suggested by Korolova [2010] [1] asks advertising systems to only use the public user information. While this way can defend the user privacy against almost all the attacks, it is highly unlikely that any for-profit corporation will use this way because it would make targeted advertising campaigns unfeasible. Lindell and Omri [2011] have argued that differential privacy is the most relevant method to provide user privacy in online advertising. Since online advertising is one of the most important application of the bandit problem, it behooves the community to devise

[1] We only discuss this way since the other suggested way is specific to Facebook.

differentially-private bandit algorithms. In the next section we take a look at such algorithms.

10.2.2 Previous work

The customary approach to achieve differential privacy in bandits is to employ a differentially private mechanism on the user feedback. This approach ensures differential privacy as stated by the following proposition [Proposition 2.1 from Dwork and Roth [2014]]

Proposition 10.1. *Let $\mathcal{M} : Domain(\mathcal{M}) \to R$ be a (ϵ, δ)-differentially private mechanism. Let $f : R \to R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : Domain(\mathcal{M}) \to R'$ is (ϵ, δ)-differentially private.*

Depending upon where the (ϵ, δ)-differentially private mechanism is placed, we classify the differential privacy algorithms considered in the community into two settings. In the first setting, the (ϵ, δ)-differentially private mechanism is located at the learner's end. The learner receives the true feedback, employs the differential privacy mechanism and runs the bandit algorithm on the output of the private mechanism. Crucially, the learner has access to true feedback. We shall call this *privacy preserving output*. In the second setting, the (ϵ, δ)-differentially private mechanism is located outside learner's control. The learner receives the differentially private input. We call this as *privacy preserving input*.

To the best of our knowledge, hitherto all the previous works applying differential privacy on bandits have used the setting of privacy preserving output. We briefly summarize some of the salient work below.

Differentially private UCB Sampling

Mishra and Thakurta [2015] provide a high-probability differentially private algorithm for the stochastic MAB problem. The proposed algorithm, called Differentially private UCB Sampling, makes use of *Tree based aggregation* as a differential privacy mechanism. Tree based aggregation, proposed by Chan et al. [2011] and Dwork et al. [2009], is an effective way of releasing private continual statistics over an input database that is dynamic and evolving over time.

Algorithm 25 Differentially private UCB Sampling

Input: Time horizon T, set of arms $A = \{1, \ldots, K\}$
Parameters: Privacy parameter ϵ, failure probability γ

Create an empty tree Tree_a with T leaves for each arm a. Set $\epsilon_0 \leftarrow \epsilon/K$.
for $t = 1, \ldots, K$ **do**
    Pull arm a_t and observe reward $x_{a_t}(t)$.
    Insert $x_{a_t}(t)$ into Tree_{a_t} via *tree based aggregation* with privacy parameter ϵ_0.
    Number of pulls $N_{a_t}(t) = 1$
end for
Confidence relaxation $\Gamma \leftarrow \frac{K \log^2 T \log((KT \log T)/\gamma)}{\epsilon}$.
for $t \leftarrow K + 1, \ldots T$ **do**
    $\mathsf{TotalReward}_a(t) \leftarrow$ Total reward computed using Tree_a for all $a \in A$.
    Pull arm $a_t = \text{argmax}_{a \in A} \left(\frac{\mathsf{TotalReward}_a(t)}{N_a(t)} + \sqrt{\frac{2 \log t}{N_a(t)}} + \frac{\Gamma}{N_a(t)} \right)$
    Insert $x_{a_t}(t)$ into Tree_{a_t} using *tree based aggregation* with privacy parameter ϵ_0.
end for

Differentially private UCB Sampling uses a tree for each of the K arms to create a (ϵ/K)-differentially private reward sequence for each arm. The empirical mean is computed using these differentially-private reward sequences. To counter the noise added to the empirical mean of the arm rewards, a normalised confidence relaxation term is added to the upper confidence bound of every arm. With the probability of $1 - \gamma$, the expected regret of this algorithm is

$$O\left(\sum_{a \in A: \mu_a < mu_*} \frac{K \log^2 T \log(KT/\gamma)}{\epsilon \Delta_a} + \Delta_a \right)$$

where μ_a is the mean of the rewards for arm a, μ_* is the optimal mean and $\Delta_a = \mu_* - \mu_a$.

DP-UCB-BOUND and DP-UCB-INT

Tossou and Dimitrakakis [2016] too provide upper confidence bound based differentially private algorithms for the stochastic MAB problem. The algorithm DP-UCB-BOUND (26)uses hybrid noise (Chan et al. [2011]) as differential privacy mechanism. Hybrid noise combines logarithmic and binary noise at regular intervals. The empirical mean reward of each arm is computed is using the hybrid mechanism used for the respective arm. A suitable term to account for the uncertainty due to privacy is also added besides the usual confidence interval to compute the optimistic estimate for each arm.

Algorithm 26 DP-UCB-BOUND

Input: Time horizon T, set of arms $A = \{1, \dots, K\}$
Instantiate K hybrid mechanisms, one for each arm.
$N_a \leftarrow 0 \quad \forall a \in A$
for $t = 1 \dots T$ **do**
  if $t \leq K$ **then**
    Play arm $a = t$ observe $x_a(t)$.
    Insert $x_a(t)$ to the hybrid mechanism for arm a.
    $N_a = N_a + 1$.
  else
    for $a \in A$ **do**
      $s_a(t) \leftarrow$ total sum computed using the hybrid mechanism for arm a.
      if N_a is a power of 2 **then**
        $v_a \leftarrow \frac{\sqrt{8}}{\epsilon} \log(4t^4)$
      else
        $v_a \leftarrow \frac{\sqrt{8}}{\epsilon} \log(4t^4) \log(N_a) + \frac{\sqrt{8}}{\epsilon} \log(4t^4)$
      end if
    end for
    Pull arm $a_t = \text{argmax}_a \frac{s_a(t)}{N_a} + \sqrt{\frac{2 \log t}{N_a}} + \frac{v_a}{N_a}$ observe $x_{a_t}(t)$.
    Insert $x_{a_t}(t)$ to the hybrid mechanism for arm a_t.
    $N_{a_t} = N_{a_t} + 1$.
  end if
end for

Algorithm 27 DP-UCB-INT

Input: Time horizon T, set of arms $A = \{1, \dots, K\}$
Parameters: $\epsilon \in (0, 1]$, $v \in (1, 1.5]$; privacy rate.
$f \leftarrow \lceil \frac{1}{\epsilon} \rceil$, $Mean_a \leftarrow 0, s_a \leftarrow 0, N_a \leftarrow 0 \quad \forall a \in A$
for $t \leftarrow 1, \dots, T$ **do**
  if $t \leq Kf$ **then**
    play arm $a = (t-1) \mod K + 1$ and observe $x_a(t)$
    $s_a \leftarrow s_a + x_a(t)$, $N_a = N_a + 1$
  else
    for $a \in A$ **do**
      if $N_a \mod f = 0$ **then**
        $Mean_a \leftarrow \frac{s_a}{N_a} + Lap\left(0, \frac{1}{N_a^{1-v/2}}\right) + \sqrt{\frac{2 \log t}{N_a}}$
      end if
    end for
    Pull arm $a_t = \text{argmax}_a \, Mean_a$ and observe $x_a(t)$
    $s_{a_t t} \leftarrow s_{a_t} + x_{a_t}(t)$, $N_{a_t} = N_{a_t} + 1$
  end if
end for

The algorithm DP-UCB-INT (27) uses Laplace noise to achieve differential privacy. The Laplace noise is computed using a privacy parameter accepted by the algorithm. The expected cumulative regret of DP-UCB-INT is upper bounded by

$$\sum_{a:\mu_a<\mu_*} \Delta_a \left[\frac{1}{\epsilon} + \frac{8}{\Delta_a^2} \log T + 4\zeta(1.5) \right]$$

where μ_a is the mean of the rewards for arm a, μ_* is the optimal mean, $\Delta_a = \mu_* - \mu_a$ and ζ denotes the Riemann zeta function.

Having seen the relevant past work in this topic, let us see how the setting of corrupt bandits can be utilised to achieve privacy preserving input.

10.3 Corrupt bandits for differential privacy

Following the conventional way, we too shall achieve differential privacy by employing a differentially private mechanism on the user feedback. Our solution for differential privacy differs from the previous work on differential privacy in bandits due to the placement of the differentially private mechanism. In our solution, the differential private mechanism is placed outside learner's control unlike the previous work. Thus, according to the classification introduced earlier, our solution provides privacy preserving input whereas the previous work in bandits provided privacy preserving output.

To understand the motivation for privacy preserving input, let us consider these settings in the context of Internet advertising. An advertising system receives, as input, feedback from the users which includes private information about them. The advertising system employs a suitable bandit algorithm and selects the ads for the users tailored to the feedback given by them. These selected ads are then given to the advertisers as the output. In the setting of privacy preserving output, privacy is maintained between the advertising system and the advertisers. In the parlance of differential privacy, the advertising system is the data curator and the the advertiser is the data receiver. Recall from section that, in private data analysis which is the goal of differential privacy, the legitimate receiver is the same as the snooping adversary and hence the receiver does not have access to privacy-compromising information.

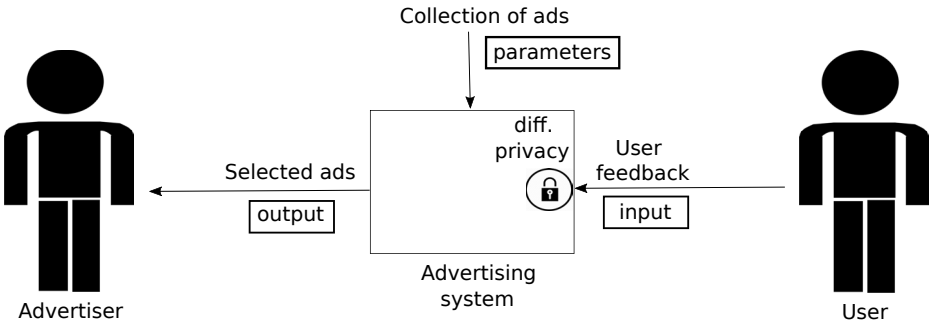


FIGURE 10.1: Internet advertising with privacy preserving output

Hence if the setting of privacy preserving output is used for the application of Internet advertising, personal user information is protected from the advertisers. This is illustrated in figure 10.1

Typically, advertising systems are established by leading social networks, web browsers and other popular websites. Harper [2010] claim that “most websites and ad networks do not ‘sell’ information about their users. In targeted online advertising, the business model is to sell space to advertisers - giving them access to people (“eyeballs") based on their demographics and interests. If an ad network sold personal and contact info, it would undercut its advertising business and its own profitability." However, despite the best of intentions by the corporations hosting the advertising systems, personal user information stored at the advertising systems can nonetheless be misused for malicious purpose. Indeed, Korolova [2010] note that using Facebook’s advertising systems one can infer user age or sexual orientation, relationship status, political and religious affiliation, presence or absence of a particular interest, as well as exact birthday. Kosinski et al. [2013] show that it is possible to accurately predict a range of highly sensitive personal attributes including:

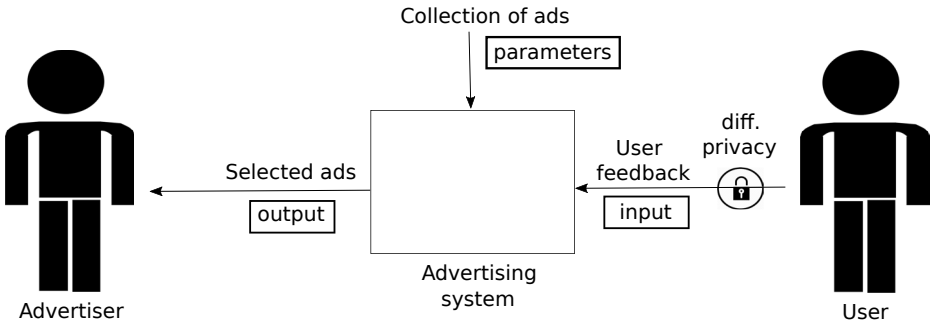


FIGURE 10.2: Internet advertising with privacy preserving input

sexual orientation, ethnicity, religious and political views, personality traits, intelligence, happiness, use of addictive substances, parental separation, age, and gender from easily accessible digital records of behaviour. Such possible breach of privacy necessitates us to protect personal user information not only from the advertisers but also from the advertising systems. The setting of privacy preserving input is able to achieve this goal unlike the setting of privacy preserving output. The setting of privacy preserving used in the application of Internet advertising is illustrated in the figure 10.2.

Privacy preserving input has been used for data collection by Wang et al. [2016]. They used randomized response to perturb sensitive information before being collected by an untrusted server so as to limit the server's ability to learn with confidence the sensitive information. We too shall use the corruption process as a mechanism to provide differentially privacy. We next define a bandit feedback corruption scheme able to provide (ϵ, δ)-differentially privacy.

Definition 10.2. *((ϵ, δ)-differentially private bandit feedback corruption scheme) A bandit feedback corruption scheme $\tilde{g}$ is (ϵ, δ)-differentially private if for all reward sequences*

$R_{t1}, \ldots, R_{t2}$ and $R'_{t1} \ldots, R'_{t2}$ that differ in at most one reward, and for all $\mathcal{S} \subseteq Range(g)$

$$\mathbb{P}[\tilde{g}(R_{t1}, \ldots, R_{t2}) \in \mathcal{S}] \leq e^{\epsilon} \cdot \mathbb{P}[\tilde{g}(R'_{t1}, \ldots, R'_{t2}) \in \mathcal{S}] + \delta$$

If $\delta = 0$, then $\tilde{g}$ is said to be ϵ-differentially private.

In the case, where corruption is done by randomized response, differential privacy requires that

$$\max_{1 \leq a \leq K} \left(\frac{p_{00}(a)}{p_{11}(a)}, \frac{p_{11}(a)}{p_{10}(a)} \right) \leq e^{\epsilon} + \delta$$

By ensuring the appropriate values for the parameters of randomized response, users can can send differentially private feedback to the learner. The learner can then use kl-UCB-CF or TS-CF to learn from such feedback. Since the algorithms can only access differentially private input, using proposition 10.1 it follows that the above process achieves differential privacy. In the next theorem, we provide an upper bound on the resulting cumulative regret.

Corollary 10.1. *The expected regret of* kl-UCB-CF *or* TS-CF *with (ϵ, δ)-differentially private bandit feedback corruption scheme is $\mathcal{O}\Big(\sum_{a=2}^{K} \left(\frac{e^{\epsilon}+\delta+1}{e^{\epsilon}+\delta-1} \right)^2 \frac{\log(T)}{\Delta_a} \Big)$.*

Proof. From corollary 9.2, we can see that to achieve lower expected regret, $p_{00}(a) + p_{11}$(a) is to be maximized for all $1 \leq a \leq K$. Using result 1 from [Wang et al., 2016, p. 3], we can state that, in order to achieve (ϵ, δ)-differential privacy while maximizing $p_{00}(a) + p_{11}(a)$,

$$\mathbb{M}_a := \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} \frac{e^{\epsilon}+\delta}{1+e^{\epsilon}+\delta} & \frac{1}{1+e^{\epsilon}+\delta} \\ \frac{1}{1+e^{\epsilon}+\delta} & \frac{e^{\epsilon}+\delta}{1+e^{\epsilon}+\delta} \end{bmatrix} \end{array} \tag{10.1}$$

Substituting the values of $p_{00}(a)$ and $p_{11}(a)$ in the bound given by corollary 9.2 completes the proof. □

In the next chapter, we provide the results which show the performance of kl-UCB-CFand TS-CFon various experiments.

Chapter 11

Empirical Evaluation

This chapter provides the results of the performance of KLUCBCF and TS-CF on various experiments. In Section 11.1, we compare the performance of kl-UCB-CF and TS-CFagainst that of the baseline algorithm which we shall introduce shortly. In Section 11.2.1, we compare the performances of KLUCBCF and TS-CF over a period of time. In Section 11.2.2, we compare them over varying corruption parameters. Lastly, in Section 11.2.3, we examine the effect on regret while these algorithms are used to provide certain levels of differential privacy.

Before delving into the experiments, we first consider a naive algorithm which we call WRAPPER to be used as a baseline. The WRAPPER algorithm simply applies the appropriate inverse corruption function to the empirical feedback values and uses the result as a substitute for empirical reward. It then treats the corrupt bandit problem as a classical MAB problem and solves it using of any classical MAB algorithm as a black-box. Let `CBA` denote the MAB algorithm used. The `CBA` has three subroutines: `init`(), `decide`() and `feedback`(). The `init`() subroutine simply clears its state. The `decide`() subroutine returns the next arm to play and the `set_feedback`() subroutine provides the feedback to the algorithm.

Algorithm 28 WRAPPER for MAB with corrupted feedback (kl-UCB-CF

Input: A bandit model with a set of arms $A \coloneqq \{1, \ldots, K\}$ arms with unknown reward means $\mu_1, \ldots, \mu_K$ and unknown feedback means $\lambda_1, \ldots, \lambda_K$ and, strictly monotonic and continuous corruption functions $g_1, \ldots, g_k$

1: `CBA.init()`
2: **for** $t \leftarrow 1, \ldots$ **do**
3: $a \leftarrow$ `CBA.decide()`
4: Pull arm a and observe feedback $y_a(t)$.
5: `CBA.set_feedback(`$\mathtt{g_a^{-1}(y_a(t))}$`)`
6: **end for**

It is easy too see that this naive algorithm won't work for the corruption functions in which $\mathbb{E}(g_{\cdot}^{-1}(y)) \neq g_{\cdot}^{-1}(\mathbb{E}(y))$. Even while using corruption functions such that $\mathbb{E}(g_{\cdot}^{-1}(y)) = g_{\cdot}^{-1}(\mathbb{E}(y))$, this algorithm gives worse performance than the sophisticated algorithms provided below. This can be verified below. The inferior performance of WRAPPER is because it doesn't take into account the variance of the sequence generated by applying inverse corruption function to the empirical feedback values.

For our experiments, we use 6 scenarios with Bernoulli distributions for arm rewards. The mean arm rewards for all the considered scenarios are enlisted in table 11.1.

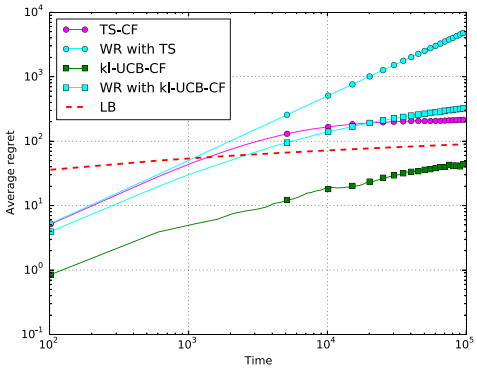
TABLE 11.1: Bernoulli mean arm rewards for experimental scenarios.

Scenario	Arms									
	1	2	3	4	5	6	7	8	9	10
1	0.55	0.45								
2	0.9	0.6								
3	0.9	0.8								
4	0.9	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
5	0.9	0.8	0.8	0.8	0.7	0.7	0.7	0.6	0.6	0.6
6	0.9	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

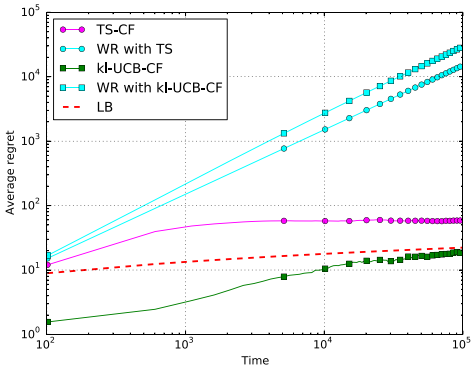
Randomized response was employed to corrupt the feedback in all the experiments. First we demonstrate the performance superiority of kl-UCB-CF and TS-CF over the naive WRAPPER algorithm.

11.1 Comparison of dedicated corrupt bandit algorithms with WRAPPER

We compare the performance of kl-UCB-CF and TS-CF against the instantiations of the WRAPPER algorithm with given kl-UCB and TS given as blackbox subroutines respectively. For the optimal arm, $p_{00} = p_{11} = 0.6$ and for all the other arms, both p_{00} and p_{11} were set to 0.9. Each experiment was repeated 1000 times. Figures 11.1, 11.2 and 11.3 show the average regret plots for kl-UCB-CF , TS-CF and their corresponding WRAPPER instantiations.

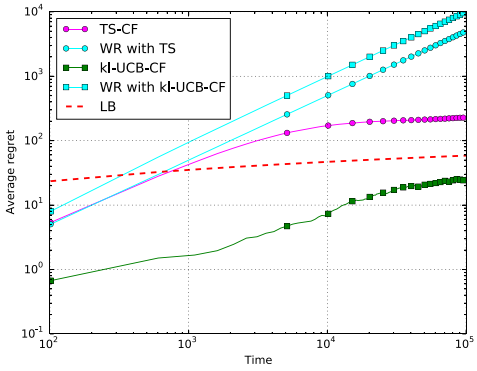


(A) Regret plots for scenario 1

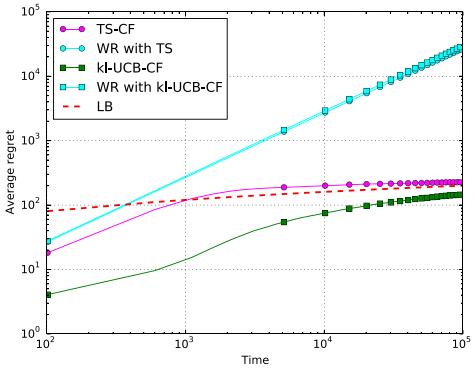


(B) Regret plots for scenario 2

FIGURE 11.1: Regret plots comparing dedicated corrupt bandit algorithms with WRAPPER

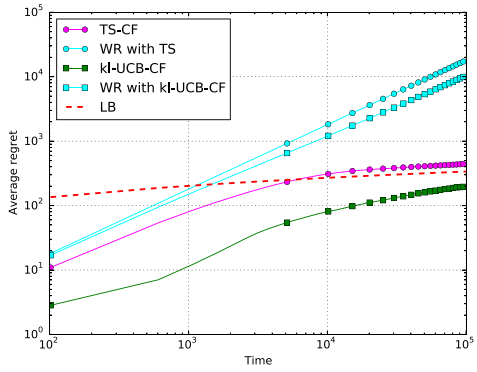


(A) Regret plots for scenario 3

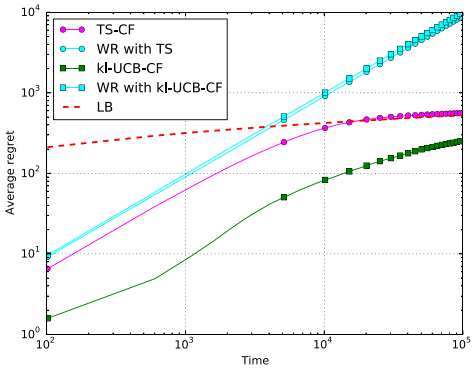


(B) Regret plots for scenario 4

FIGURE 11.2: Regret plots comparing dedicated corrupt bandit algorithms with WRAPPER



(A) Regret plots for scenario 5



(B) Regret plots for scenario 6

FIGURE 11.3: Regret plots comparing dedicated corrupt bandit algorithms with WRAPPER

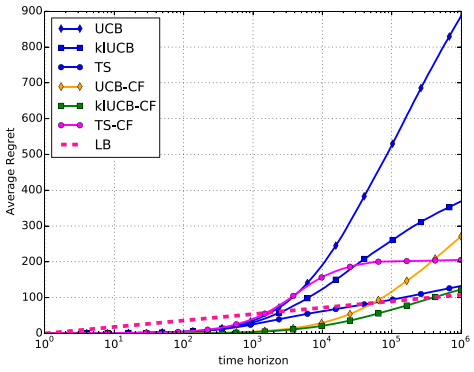
11.2 Comparison between kl-UCB-CF, UCB-CF and TS-CF

We now demonstrate the empirical performance of kl-UCB-CF, UCB-CF and TS-CF. To demonstrate the inability of the bandit algorithms to deal with corrupted feedback, we also consider kl-UCB, UCB1, and TS. These algorithms solve the corrupted bandit problem viewing the feedback as true rewards as is the assumption for classical MAB problem.

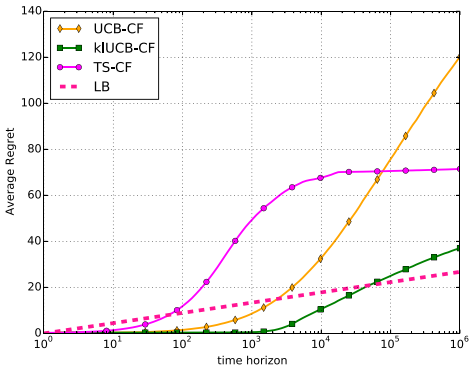
Note that, when all the arms are corrupted by the same increasing function, kl-UCB-CF, UCB-CF, and TS-CF are equivalent to kl-UCB, UCB1, and TS, respectively. We investigate more complex situations having varying level of corruption across the arms, for which the classical bandit algorithms may perform poorly. We give the plots for each of the 6 scenarios for three kinds of comparisons: comparison over time, comparison with varying corruption paramters and comparions with varying level of differetial privacy.

11.2.1 Comparison over a period of time

Here, we aim to see the effect of time over the cumulative regret of the considered algorithms. We keep the corruption parameters same across time. For the optimal arm, $p_{00} = p_{11} = 0.6$ and for all the other arms, both p_{00} and p_{11} were set to 0.9. We vary time to 10^6 Each experiment was repeated 1000 times and the average regret is plotted against time in Figures 11.4, 11.5 and 11.6 for all the scenarios given in the Table 11.1. In Figure 11.4a, we include the regret curves for the traditional bandit algorithms (treating feedback as reward) as well to illustrate their comparative performance. The performance superiority of the dedicated algorithms for corrupt bandits is more pronounced in the scenarios in which the corruption causes the best arm to be switched (or at least to be no longer unique). Hence, to have more legible figures we only display the regret curves for kl-UCB-CF TS-CF and UCB-CF for the rest of the scenarios.

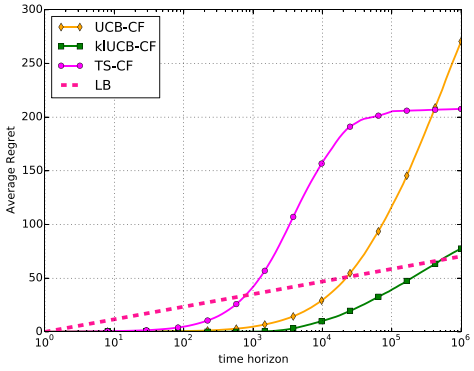


(A) Regret plots for scenario 1

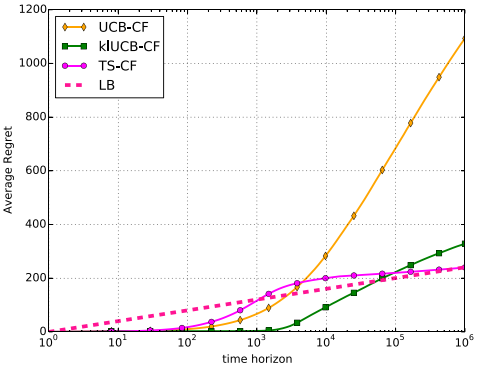


(B) Regret plots for scenario 2

FIGURE 11.4: Regret plots for comparison over a period of time

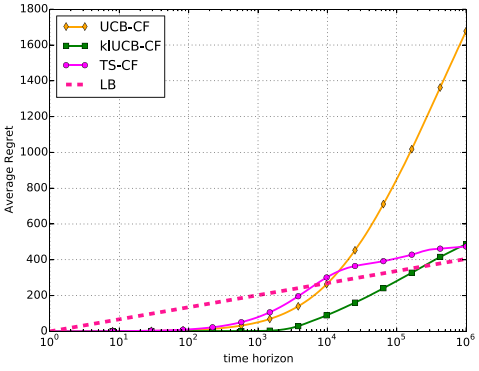


(A) Regret plots for scenario 3

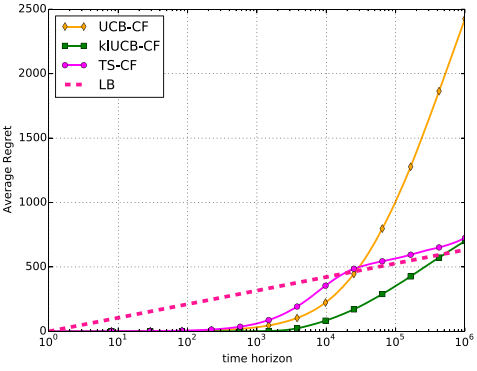


(B) Regret plots for scenario 4

FIGURE 11.5: Regret plots for comparison over a period of time



(A) Regret plots for scenario 5



(B) Regret plots for scenario 6

FIGURE 11.6: Regret plots for comparison over a period of time

11.2.2 Comparison with varying corruption parameters

Here, we aim to see the effect of corruption over the cumulative regret of the considered algorithms. We fix the time horizon at 10^5 for all the comparisons. For all the arms, the corruption parameters ($p = p_{00} = p_{11}$) vary from 0 to 1. Each experiment was repeated 1000 times and the average cumulative regret was plotted against the value of corruption parameters in Figures 11.7, 11.8 and 11.9 for all the scenarios given in the Table 11.1.

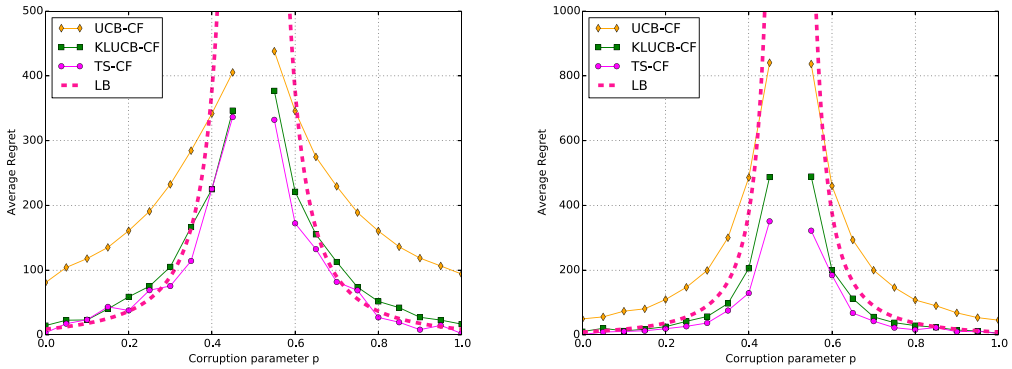
These experiments also provide an opportunity to test a boundary case where p_{00}= p_{11} = 0.5. Recall from Theorem 8.2 that the lower bound on the cumulative regret for any uniformly efficient algorithm adheres to the following equation

$$\liminf_{T\to\infty} \frac{\mathrm{CRegret}_T(\nu)}{\log(T)} \geq \sum_{a=2}^{K} \frac{\Delta_a}{d\left(\lambda_a, g_a(\mu_1)\right)}$$

where $d(x, y) = \mathrm{KL}(\mathcal{B}(x), \mathcal{B}(y))$. When p_{00}= p_{11} = 0.5 for all the arms, $\lambda_a = g_a(\mu_1)$ for every arm a since,

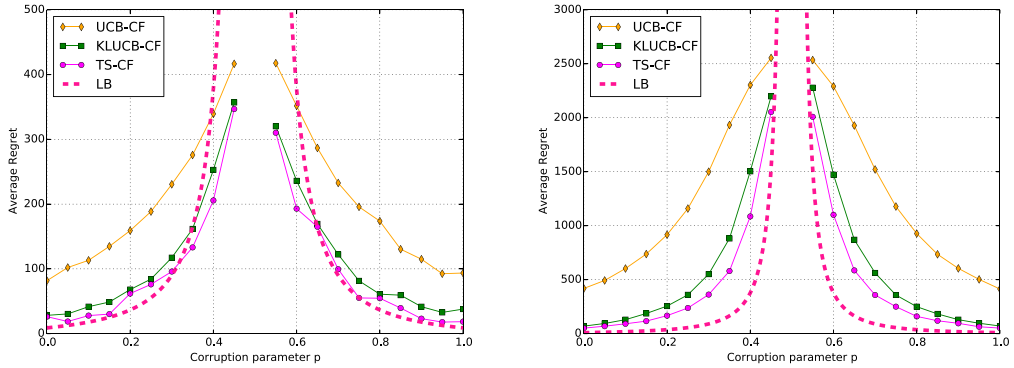
$$\lambda_a = 0.5(1-\mu_a) + 0.5\mu_a = 0.5 \quad \text{and} \quad g_a(\mu_1) = 0.5(1-\mu_1) + 0.5\mu_1 = 0.5$$

So as $d\left(\lambda_a, g_a(\mu_1)\right)$ is 0 in this case, the lower bound is undefined. Intuitively the undefined value for the lower bound at this boundary case follows from the fact that all the discriminating information for the arms is lost since the mean feedback for all of them is the same i.e. 0.5. Hence no uniformly efficient algorithm is able to learn in this case.



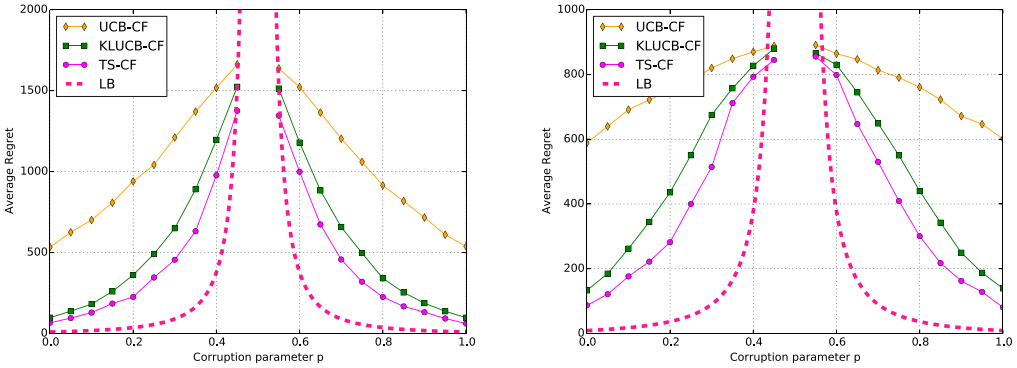
(A) Regret plots for scenario 1 (B) Regret plots for scenario 2

FIGURE 11.7: Regret plots with varying corruption parameters



(A) Regret plots for scenario 3 (B) Regret plots for scenario 4

FIGURE 11.8: Regret plots with varying corruption parameters



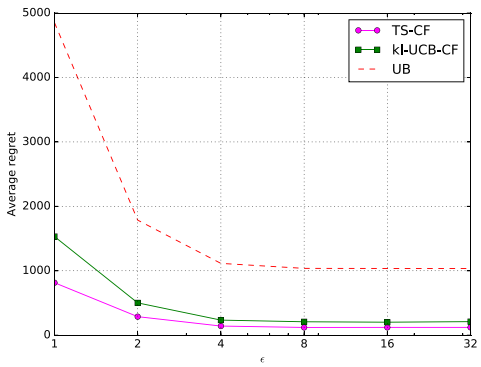
(A) Regret plots for scenario 5 (B) Regret plots for scenario 6

FIGURE 11.9: Regret plots with varying corruption parameters

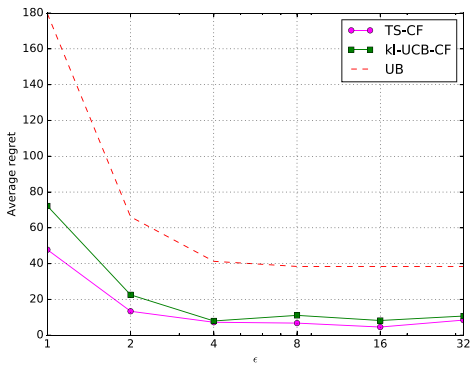
11.2.3 Comparison with varying level of differential privacy

We vary the differential privacy parameters and examine the effect on the regret of kl-UCB-CF and TS-CF. We kept $\delta = 0$ and chose ϵ from the set $\{1, 2, 4, 8, 16, 32\}$. The corruption parameters are set by substituting the values of ϵ and δ in Equation (10.1). The time horizon was fixed to 10^5 and the experiment was repeated 1000 times. The corresponding curves for average regret can be seen in Figures 11.10, 11.11 and 11.12. UB indicates the upper bound given by Corollary 10.1.

The regret for both the algorithms decreases with increasing ϵ. This behavior is expected since, lower the value of ϵ, more stringent is the level of differential privacy. The regret decreases rapidly initially as an increase in ϵ leads to massive drop in the imposed level of differential privacy and consequently the corruption parameters set

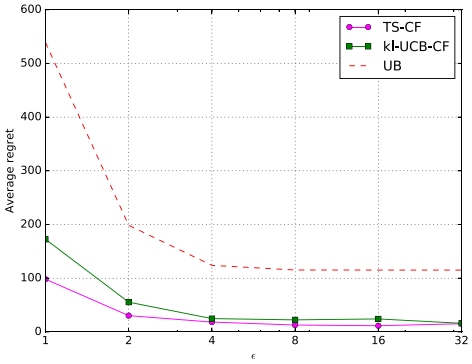


(A) Regret plots for scenario 1

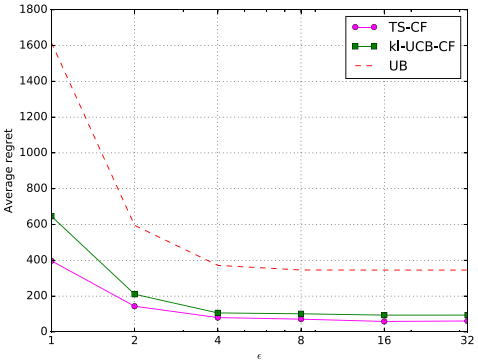


(B) Regret plots for scenario 2

FIGURE 11.10: Regret plots with varying level of differential privacy

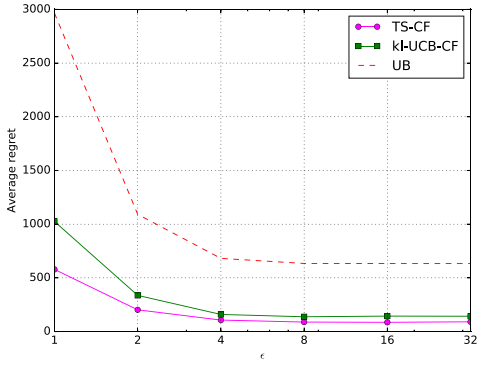


(A) Regret plots for scenario 3

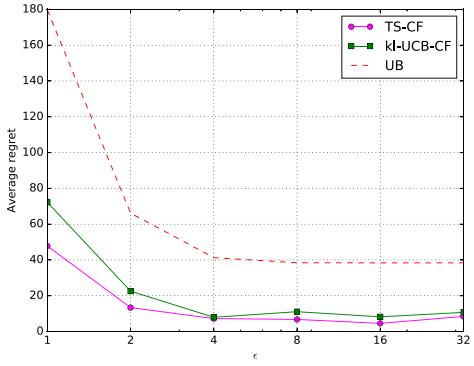


(B) Regret plots for scenario 4

FIGURE 11.11: Regret plots with varying level of differential privacy



(A) Regret plots for scenario 5



(B) Regret plots for scenario 6

FIGURE 11.12: Regret plots with varying level of differential privacy

by Equation (10.1) are adjusted. At higher values of ϵ, the regret plateaus as a change in ϵ causes an infinitesimal change in the required level of differential privacy.

Chapter 12

Corrupt Bandits as Partial Monitoring game

In this short chapter, we shall see how the Bernoulli corrupt bandits can be formulated as a partial monitoring game ((detailed in Section 1.5)).

To recapitulate, a partial monitoring game (PM) is defined by a tuple $(\boldsymbol{N}, \boldsymbol{M}, \boldsymbol{\Sigma}, \mathcal{G}, \mathcal{H})$ where $\boldsymbol{N}$, $\boldsymbol{M}$, $\boldsymbol{\Sigma}$, $\mathcal{G}$ and $\mathcal{H}$ are the action set, the outcome set, the feedback alphabet, the reward function and the feedback function respectively. To each action $I \in \boldsymbol{N}$ and outcome $J \in \boldsymbol{M}$, the *reward function* $\mathcal{G}$ associates a real-valued gain $\mathcal{G}(I, J)$ and the *feedback function* $\mathcal{H}$ associates a feedback symbol $\mathcal{H}(I, J) \in \boldsymbol{\Sigma}$.

For the binary MAB problem with corrupted feedback, a partial monitoring formulation is provided with an action set $\boldsymbol{N}$ = the set of arms, an alphabet $\boldsymbol{\Sigma} = \{0, 1\}$ and a set of environment outcomes which are vectors $\boldsymbol{m} \in \boldsymbol{M} = \boldsymbol{x} \times \boldsymbol{y}$ where $\boldsymbol{x}$ is the reward vector containing the rewards of all the K arms and $\boldsymbol{y}$ is the feedback vector containing the feedbacks of all the K arms. When the environment selects an outcome and the learner selects an arm $a \in \boldsymbol{N}$, reward and feedback are as follows:

$$\mathcal{G}(a, (\mathbf{x}, \mathbf{y})) = \mathbf{x}_a$$
$$\mathcal{H}(a, (\mathbf{x}, \mathbf{y})) = \mathbf{y}_a$$

For stochastic corrupt bandits, $\mathbb{E}\mathbf{y}_a = g_a(\mathbb{E}\mathbf{x}_a)$ and for adversarial corrupt bandits $\hat{y}_a(T) = g_a(\hat{x}_a(T))$ where $\hat{y}_a(T)$ is the mean empirical feedback and $\hat{x}_a(T)$ is the mean empirical reward at horizon T.

$$\mathcal{G} = \begin{array}{c|cccccccccccccccc} & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

$$\mathcal{H} = \begin{array}{c|cccccccccccccccc} & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

FIGURE 12.1: Gain matrix $\mathcal{G}$ and feedback matrix $\mathcal{H}$ for a 2-armed binary MAB problem with corrupted feedback resulting in 2 non-duplicate actions and 16 possible outcomes.

$$\mathcal{S}_{(1)} = \begin{array}{c|cccccccccccccccc} & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}$$

$$\mathcal{S}_{(2)} = \begin{array}{c|cccccccccccccccc} & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

FIGURE 12.2: Signal matrices for the same problem as in Figure 12.1.

To illustrate this formalism, we encode a 2-armed binary MAB problem with corrupted feedback as a PM problem in Figure 12.1. The first element of every column is of the form $\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{y}_1\boldsymbol{y}_2$ where $\boldsymbol{x}_a$ and $\boldsymbol{y}_a$ are reward and feedback for the a^{th} arm. The first element of every row is of the form d where d denotes the arm being picked by the learner. Figure 12.2 shows the signal matrix for both the actions. Recall from definition 1.5 that, the signal matrix for an action is the incidence matrix of symbols and outcomes.

As shown above, the formulation of a K-armed Bernoulli corrupt bandit problem as a partial monitoring game requires matrices of dimension $K \times 2^K$ for gain matrix and feedback matrix. Even for moderate values of K, this requirement is impractical.

This brings us to the end of this chapter. Next, we proceed to the appendices where we present the detailed proofs for various theorems stated earlier.

Appendix B

Proof for Theorem 9.1

This proof proceeds along the lines of the proof for the sample complexity for the median elimination algorithm given in Even-Dar et al. [2006]. Acorrding to our assumption, $g_.$ is $(1/\sigma)$-Lipschitz i.e.

$$|g_.(v_1) - g_.(v_2)| \leq (v_1 - v_2)/\sigma \tag{B.1}$$

Furthermore, since $g_.$ is differentiable and $(1/\sigma)$-Lipschitz, $g_.^{-1}$ is σ-Lipschitz

$$|g_.^{-1}(v_1) - g_.^{-1}(v_2)| \leq \sigma(v_1 - v_2) \tag{B.2}$$

First we provide a Lemma which states that the expected reward of the best arm in S_l drops by at most ϵ.

Lemma B.1. *For every phase l,*

$$\mathbb{P}[\max_{a \in S_l} \mu_a \leq \max_{b \in S_{l+1}} \mu_b + \epsilon_l] \geq 1 - \delta_l$$

Proof. Without loss of generality, let us assume $l = 1$ while proving this lemma. Since $S_1 = [K]$, the best arm in S_1 is in fact the optimal arm 1. Let E_1 be the event that during the first round, the empirical estimate of the best arm is pessimistic.

$$E_1 := \hat{f}_1^1 < \mu_1 - \epsilon_1/2 \equiv g_1^{-1}(\hat{\lambda}_1^1) < g_1^{-1}(\lambda_1) - \epsilon_1/2$$

Using (B.2), we can write that,

$$|g_1^{-1}(\lambda_1) - g_1^{-1}(\hat{\lambda}_1^1)| \leq \sigma(\lambda_1 - \hat{\lambda}_1^1)$$

Therefore E_1 is equivalent to

$$E_1 : \lambda_1 - \hat{\lambda}_1^1 > \epsilon_1/(2\sigma)$$

Using Hoeffding's bound, we have

$$\mathbb{P}[E_1] \leq \delta_1/3 \tag{B.3}$$

Let a be an arm which is not ϵ_1 optimal i.e.

$$\mu_1 - \mu_a > \epsilon_1 \tag{B.4}$$

let us assume that event E_1 does not hold.

$$\begin{aligned}
\neg E_1 &\equiv \hat{f}_1^1 \geq \mu_1 - \epsilon_1/2 \\
&\equiv \hat{f}_1^1 > \mu_a + \epsilon_1/2 \qquad \text{(using (B.4))}
\end{aligned}$$

The probability of the arm a being empirically better than the optimal arm when event E_1 does not hold is given by

$$\begin{aligned}
\mathbb{P}[\hat{f}_a^1 > \hat{f}_1^1 \mid \neg E_1] &= \mathbb{P}[\hat{f}_a^1 > \mu_a + \epsilon_1/2 \mid \neg E_1] \\
&\leq \mathbb{P}[\hat{\lambda}_a^1 - \lambda_a > \epsilon_1/(2\sigma) \mid \neg E_1] \quad \text{(using B.2)} \\
&\leq \delta_1/3 \quad \text{(using Hoeffding's bound)}
\end{aligned}$$

Let $\#\text{bad}$ be the number of arms that are not ϵ_1-optimal but are empirically better than the best arm.

$$\mathbb{E}[\#\text{bad} \mid \neg E_1] \leq K\delta_1/3$$

$$\mathbb{P}[\#\text{bad} \geq K/2 \mid \neg E_1] \leq \frac{K\delta_1/3}{K/2} = 2\delta_1/3 \quad \text{using Markov's inequality} \tag{B.5}$$

Therefore the failure probability i.e. the probability that every arm bring selected for round 2 is not ϵ_1-optimal which is equivalent to $\mathbb{P}[\#\text{bad} \geq K/2]$ is bounded by δ. $\square$

Lemma B.2. *The sample complexity of* ME-CF *is* $O\left((K\sigma^2/\epsilon^2)\log(1/\delta)\right)$

Proof. Let $n_l = |S_l|$. Initially we have $n_1 = K, \epsilon_1 = \epsilon/4$ and $\delta_1 = \delta/2$. Then,

$$\begin{aligned} n_l &= n_{l-1}/2 = K/2^{l-1} \\ \epsilon_l &= \frac{3}{4}\epsilon_{l-1} = \left(\frac{3}{4}\right)^{l-1}\epsilon/4 \qquad \text{and,} \\ \delta_l &= \delta_{l-1}/2 = \delta/2^l \end{aligned}$$

The total number of arm samples are given by,

$$\begin{aligned} \sum_{l=1}^{\log_2(K)} \frac{n_l \log(3/\delta_l)}{(\epsilon_l/2\sigma)^2} &= 4\sigma^2 \sum_{l=1}^{\log_2(K)} \frac{K/2^{l-1}\log(2^l 3/\delta)}{((\frac{3}{4})^{l-1}\epsilon/4)^2} \\ &= 64\sigma^2 \sum_{l=1}^{\log_2(K)} K\left(\frac{8}{9}\right)^{l-1}\left(\frac{\log(1/\delta)+\log(3)+l\log(2)}{\epsilon^2}\right) \\ &\leq 64\frac{K\sigma^2\log(1/\delta)}{\epsilon^2}\sum_{l=1}^{\infty}\left(\frac{8}{9}\right)^{l-1}(lC+C') \\ &= O\left(\frac{K\sigma^2}{\epsilon^2}log\left(\frac{1}{\delta}\right)\right) \end{aligned}$$

□

Now we can prove Theorem 9.1.

Proof. From Lemma B.1, we have that during round l, the optimal reward of the surviving arms is reduced by at-most ϵ_l with failure probability δ_l. Therefore the total error is bounded by $\sum_{l=1}^{\log_2 K}\epsilon_l = \epsilon$ and the probability of failure is bounded by $\sum_{l=1}^{\log_2 K}\delta_l = \delta$. Therefore 21 is (ϵ,δ)-PAC. By Lemma B.2, we have that the sample complexity of 21 is bounded by $O\left(\frac{K\sigma^2}{\epsilon^2}log\left(\frac{1}{\delta}\right)\right)$ □

Appendix C

Proof for Theorem 9.2

This proof closely follows the proof for the upper bound on the sample complexity of exponential-gap elimination given in Karnin et al. [2013]

The following lemma proves that with high probability, the best arm is not eliminated by the algorithm EGE-CF.

Lemma C.1. *For any round l, $\hat{f}_1^l \geq \hat{f}_{a_l}^l - \epsilon_l$, with probability at least $1 - \delta/5$.*

Proof. Consider the event $|\hat{f}_a^l - \mu_a| \geq \epsilon_l/2$. Using (B.2), $|\hat{f}_a^l - \mu_a| \geq \epsilon_l/2 \equiv \hat{\lambda}_a^l - \lambda_a \geq \epsilon_l/(2\sigma)$.

$$\begin{aligned} \therefore \mathbb{P}[|\hat{f}_a^l - \mu_a| \geq \epsilon_l/2] &= \mathbb{P}[\hat{\lambda}_a^l - \lambda_a \geq \epsilon_l/(2\sigma)] \\ &\leq exp(-2(\epsilon_l/2\sigma)^2(2/(\epsilon_l/\sigma)^2)\log(1/\delta_l)) \\ &= \delta_l \end{aligned} \tag{C.1}$$

Assume that, the optimal arm was not eliminated till the end of round $l-1$. Consider the arm a_l returned by the ME-CF in round l. Since 1 is the optimal arm, $\mu_{a_l} \leq \mu_1$ Using (C.1),

$$\begin{aligned} &\mathbb{P}[\hat{f}_{a_l}^l < \mu_{a_l} + \epsilon_l/2] \geq 1 - \delta_l \\ &\mathbb{P}[\hat{f}_{a_l}^l < \mu_1 + \epsilon_l/2] \geq 1 - \delta_l \quad \because \mu_{a_l} \leq \mu_1 \\ &\mathbb{P}[\hat{f}_{a_l}^l - \epsilon_l/2 \geq \mu_1] < \delta_l \end{aligned} \tag{C.2}$$

Using (C.1), we can also write that,

$$\mathbb{P}[\mu_1 < \hat{f}_1^l + \epsilon_l/2] \geq 1 - \delta_l$$

$$\mathbb{P}[\mu_1 \geq \hat{f}_1^l + \epsilon_l/2] < \delta_l$$
$$\mathbb{P}[\hat{f}_{a_l}^l - \epsilon_l/2 \geq \hat{f}_1^l + \epsilon_l/2] < 2\delta_l \quad \text{using (C.2)}$$
$$\mathbb{P}[\hat{f}_1^l < \hat{f}_{a_l}^l - \epsilon_l] < 2\delta_l$$

Using union bound, we can bound the error probability as follows,

$$\sum_{l=1}^{\infty} 2\delta_l \leq \sum_{l=1}^{\infty} 2\delta/(50l^2) \leq \delta/5$$

□

Recall that $\Delta_a = \mu_1 - \mu_a$. For all $0 \leq s \leq \left\lceil \log_2 \frac{1}{\min_{a=1}^{K} \Delta_a} \right\rceil$,let us define A_s as follows:

$$A_s := \{a \in [K] : 2^{-s} \leq \Delta_a \leq 2^{-s+1}\} \tag{C.3}$$

We denote the set of arms from A_s surviving after round l by $S_{l,s} = S_l \cap A_s$ for all $l, s \geq 0$.

The following lemma proves that from round s onwards, a constant fraction of the surviving arms of the set A_s is eliminated in each round.

Lemma C.2. *Assume that the optimal arm is not eliminated by* EGE-CF *till the start of round l, then for all $1 \leq s \leq l$,*

$$\mathbb{P}\left[|S_{l+1,s}| \leq \frac{1}{8}|S_{l,s}|\right] \geq 1 - 4\delta/5$$

Proof. Consider that the optimal arm reached round l. From Theorem 9.1,

$$\mathbb{P}[\mu_1 - \epsilon_l/2 > \mu_{a_l}] < \delta_l$$

Using (C.1), we can write that,

$$\mathbb{P}[\mu_{a_l} \geq \hat{f}_{a_l}^l + \epsilon_l/2] \leq \delta_l$$

Combining the above two inequalities, we can write that,

$$\mathbb{P}[\hat{f}_{a_l}^l \leq \mu_1 - \epsilon_l] \leq 2\delta_l \tag{C.4}$$

Consider a round $l \geq s$ and a sub-optimal arm $a \in A_s$ with

$$\Delta_a \geq 2^{-s} \geq 2^{-l} = 4\epsilon_l \tag{C.5}$$

let us compute the probability of arm a, surviving round l i.e. $\mathbb{P}[\hat{f}_a^l \geq \hat{f}_{a_l}^l - \epsilon_l]$. For the arm a to survive round l, either empirical estimate of a has to be optimistic i.e. $\hat{f}_a^l \geq \mu_a + \epsilon_l$ or $\hat{f}_{a_l}^l \leq \mu_1 - \epsilon_1$. Since otherwise,

$$\begin{aligned}
\hat{f}_a^l &< \mu_a + \epsilon_l \\
&< \mu_1 - \Delta_a + 2\epsilon_l \\
&\leq \mu_1 - 4\epsilon_l + 2\epsilon_l \quad \text{(using (C.5))} \\
&< \hat{f}_{a_l}^l - \epsilon_l
\end{aligned}$$

Hence we can rewrite the probability of arm a, surviving round l as,

$$\begin{aligned}
\mathbb{P}[\hat{f}_a^l \geq \hat{f}_{a_l}^l - \epsilon_l] &\leq \mathbb{P}[\hat{f}_a^l \geq \mu_a + \epsilon_l] + \mathbb{P}[\hat{f}_{a_l}^l \leq \mu_1 - \epsilon_1] \\
&\leq 3\delta_l \quad \text{(using (C.4) and (C.1))}
\end{aligned} \tag{C.6}$$

Therefore,

$$\mathbb{E}\left[S_{l+1,s}\right] \leq 3\delta_l \cdot \mathbb{E}\left[S_{l,s}\right]$$

Applying Markov's inequality,

$$\mathbb{P}\left[|S_{l+1,s}| > \frac{1}{8}|S_{l,s}|\right] < \frac{3\delta_l |S_{l,s}|}{\frac{1}{8}\left[S_{l,s}\right]} = 24\delta_l$$

We can now prove the lemma by bounding the probability of failure using union bound as follows

$$\sum_{l=1}^{\infty}\sum_{s=1}^{r} 24\delta_l \leq \sum_{l=1}^{\infty} 24\delta/(50r^2) < 4\delta/5$$

□

Lemma C.3. *With probability at least $1-\delta$, the total number of times that an arm from A_s is sampled in line 6 is $O(\sigma^2 \cdot 4^s \cdot |A_s| \cdot \log(s/\delta))$ for all s.*

Proof. Let N_s denote the total number of times an arm from A_s is pulled. By lemma C.2, if the algorithm is successful, we have

$$\begin{aligned}
N_s &= \sum_{l=1}^{\infty} |A_s| \cdot \hat{n}_l \\
&\leq \sum_{l=1}^{s-1} |A_s| \cdot \hat{n}_l + \sum_{l=s}^{\infty} |S_{l,s}| \cdot \hat{n}_l \\
&\leq |A_s| \sum_{l=1}^{s-1} \hat{n}_l + |A_s| \sum_{l=0}^{\infty} \left(\frac{1}{8}\right)^{l+1} \hat{n}_{l+s} \\
&\leq |A_s| \cdot 32\sigma^2 \sum_{l=1}^{s-1} 4^l \log\left(\frac{50l^3}{\delta}\right) + |A_s| \cdot 4^{s+1}\sigma^2 \sum_{l=0}^{\infty} 2^{-l} \log\left(\frac{50(l+s)^3}{\delta}\right) \\
&= O(\sigma^2 \cdot 4^s \cdot |A_s| \cdot \log(s/\delta))
\end{aligned}$$

□

We are now ready to prove Theorem 9.2

Proof. Lemma C.1 proves that the optimal arm is never eliminated with probability at least $1 - \delta/5$ and lemma C.2 implies that all the sub-optimal arms are eventually eliminated with probability at least $1 - 4\delta/5$. Hence using union bound, the algorithm stops at some point and returns the optimal arm with probability $1 - \delta$.

Now we turn our attention to computing the sample complexity. First let us count the number of pulls by the subroutine ME-CF. Note that the invocation of ME-CF in round l results in $O\left(\frac{|S_l|\cdot\sigma^2}{(\epsilon_l/2)^2} \log\left(\frac{1}{\delta_l}\right)\right) = O(|S_l| \cdot \hat{n}_l)$. Lastly, let us compute the number of pulls by the algorithm on line 6. Lemma C.3 asserts that, if the algorithm is successful, the number of times it pulls an arm from A_s is $O(\sigma^2 \cdot 4^s \cdot |A_s| \cdot \log(s/\delta))$. From the definition of A_s(C.3), we can write that $2^s < 2/\Delta_a$ for all $a \in A_s$.

$$N_s = O(\sigma^2 \cdot 4^s \cdot |A_s| \cdot \log(s/\delta)) = O\left(\sigma^2 \sum_{a \in A_s} \frac{1}{\Delta_a^2} \log\left(\frac{1}{\delta} \log \frac{1}{\Delta_a}\right)\right)$$

By summing over all s, we obtain the sample complexity of the algorithm as

$$O\left(\sigma^2 \sum_{a=2}^{K} \frac{1}{\Delta_a^2} \log\left(\frac{1}{\delta} \log \frac{1}{\Delta_a}\right)\right)$$

□

Appendix D

Proof for Theorem 9.3

To get an upper bound on the expected regret of this algorithm, we first bound $\mathbb{E}[N_a(t)]$ for all the non-optimal arms a. $\hat{a}_t :=$ the arm pulled by the algorithm at time t. Note that, we assume 1 to be the optimal arm.

$$\mathbb{E}(N_a(T)) = 1 + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a)$$

Depending upon if g_a and g_1 are increasing or decreasing there are four possible sub-cases:

- Both g_1 and g_a are increasing.

$$\begin{aligned}
&(\hat{a}_{t+1} = a) \\
&\subseteq (u_1(t) < g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, u_1(t) \geq g_1(\mu_1)) \\
&= (u_1(t) < g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, g_1^{-1}(u_1(t)) \geq \mu_1) && \text{since } g_1^{-1} \text{ is increasing} \\
&= (u_1(t) < g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, g_a^{-1}(u_a(t)) \geq \mu_1) && \text{since } \mathrm{Index}_a > \mathrm{Index}_1 \\
&= (u_1(t) < g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1)) && \text{since } g_a \text{ is increasing}
\end{aligned}$$

$$\therefore \mathbb{E}(N_a(T)) \leq 1 + \sum_{t=K}^{T-1} \mathbb{P}(u_1(t) < g_1(\mu_1)) + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1)) \tag{D.1}$$

- g_1 is decreasing and g_a is increasing.

$$\begin{aligned}
&(\hat{a}_{t+1} = a) \\
&\subseteq (\ell_1(t) > g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, \ell_1(t) \leq g_1(\mu_1))
\end{aligned}$$

$$
\begin{aligned}
&= (\ell_1(t) > g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, g_1^{-1}(\ell_1(t)) \geq \mu_1) && \text{since } g_1 \text{ is decreasing} \\
&= (\ell_1(t) > g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, g_a^{-1}(u_a(t)) \geq \mu_1) && \text{since } \mathrm{Index}_a > \mathrm{Index}_1 \\
&= (\ell_1(t) > g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1)) && \text{since } g_a \text{ is increasing}
\end{aligned}
$$

$$
\therefore \mathbb{E}(N_a(T)) \leq 1 + \sum_{t=K}^{T-1} \mathbb{P}(\ell_1(t) > g_1(\mu_1)) + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1)) \tag{D.2}
$$

- g_1 is increasing and g_a is decreasing.

$$
\begin{aligned}
&(\hat{a}_{t+1} = a) \\
&\subseteq (u_1(t) < g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, u_1(t) \geq g_1(\mu_1)) \\
&= (u_1(t) < g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, g_1^{-1}(u_1(t)) \geq \mu_1) && \text{since } g_1 \text{ is increasing} \\
&= (u_1(t) < g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, g_a^{-1}(\ell_a(t)) \geq \mu_1) && \text{since } \mathrm{Index}_a > \mathrm{Index}_1 \\
&= (u_1(t) < g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, \ell_a(t) \leq g_a(\mu_1)) && \text{since } g_a \text{ is decreasing}
\end{aligned}
$$

$$
\therefore \mathbb{E}(N_a(T)) \leq 1 + \sum_{t=K}^{T-1} \mathbb{P}(u_1(t) < g_1(\mu_1)) + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \ell_a(t) \leq g_a(\mu_1)) \tag{D.3}
$$

- g_1 is decreasing and g_a is decreasing.

$$
\begin{aligned}
&(\hat{a}_{t+1} = a) \\
&\subseteq (\ell_1(t) > g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, \ell_1(t) \leq g_1(\mu_1)) \\
&= (\ell_1(t) > g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, g_1^{-1}(\ell_1(t)) \geq \mu_1) && \text{since } g_1 \text{ is decreasing} \\
&= (\ell_1(t) > g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, g_a^{-1}(\ell_a(t)) \geq \mu_1) && \text{since } \mathrm{Index}_a > \mathrm{Index}_1 \\
&= (\ell_1(t) > g_1(\mu_1)) \cup (\hat{a}_{t+1} = a, \ell_a(t) \leq g_a(\mu_1)) && \text{since } g_a \text{ is decreasing}
\end{aligned}
$$

$$
\therefore \mathbb{E}(N_a(T)) \leq 1 + \sum_{t=K}^{T-1} \mathbb{P}(\ell_1(t) > g_1(\mu_1)) + \sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \ell_a(t) \leq g_a(\mu_1)) \tag{D.4}
$$

Let $\hat{\lambda}_{a,s} \coloneqq$ observed mean feedback from arm a after s samples.

We first upper bound the two sums

$$\sum_{t=K}^{T-1} \mathbb{P}(u_1(t) < g_1(\mu_1)) \text{ and } \sum_{t=K}^{T-1} \mathbb{P}(\ell_1(t) > g_1(\mu_1)) \tag{D.5}$$

using that $\ell_1(t)$ and $u_1(t)$ are respectively lower and upper confidence bound on $g_1(\mu_1)$. Indeed,

$$\begin{aligned}
\mathbb{P}(u_1(t) < g_1(\mu_1)) &\leq \mathbb{P}\Big(g_1(\mu_1) > \hat{\lambda}_1(t) \text{ and } N_1(t) d(\hat{\lambda}_1(t), g_1(\mu_1)) \geq f(t)\Big) \\
&\leq \mathbb{P}\Big(\exists s \in \{1, \dots, t\} : g_1(\mu_1) > \hat{\lambda}_{1,s} \text{ and } s d(\hat{\lambda}_{1,s}, g_1(\mu_1)) \geq f(t)\Big) \\
&\leq min\{1, e\lceil f(t) \log t \rceil e^{-f(t)}\},
\end{aligned}$$

where the upper bound follows from Lemma 2 in Cappé et al. [2013], and the fact that $\hat{\lambda}_{1,s}$ is the empirical mean of s Bernoulli samples with mean $g_1(\mu_1)$. Similarly, one has

$$\mathbb{P}(\ell_1(t) > g_1(\mu_1)) \leq \min\{1, e\lceil f(t) \log t \rceil e^{-f(t)}\}.$$

As $f(t) \coloneqq \log t + 3(\log\log t)$ for $t \geq 3$,

$$e\lceil f(t) \log t \rceil \leq 4e \log^2 t,$$

the two quantities in (D.5) can be upper bounded by

$$\begin{aligned}
1 + \sum_{t=3}^{T-1} e\lceil f(t) \log t \rceil e^{-f(t)} &\leq 1 + \sum_{t=3}^{T-1} 4e \cdot \log^2 t \cdot e^{-f(t)} \\
&= 1 + 4e \sum_{t=3}^{T-1} \frac{1}{t \log t} \\
&\leq 4e\Big(\frac{1}{3\log 3} + \int_3^{T-1} \frac{1}{t \log t} dt\Big) \\
&\leq 4e\Big(\frac{1}{3\log 3} + \log\left(\log\left(T-1\right)\right) - \log\left(\log 3\right)\Big) \\
&\leq 3 + 4e \log\left(\log T\right).
\end{aligned}$$

This proves that

$$\sum_{t=K}^{T-1} \mathbb{P}(u_1(t) < g_1(\mu_1)) \quad \leq \quad 3 + 4e \log(\log T) \in o(\log T) \tag{D.6}$$

$$\sum_{t=K}^{T-1} \mathbb{P}(\ell_1(t) > g_1(\mu_1)) \quad \leq \quad 3 + 4e \log(\log T) \in o(\log T) \tag{D.7}$$

We now turn our attention to two other sums involved in the upper bounds we gave for $\mathbb{E}(N_a(t))$. We introduce the notation $d^+(x,y) = d(x,y)\mathbb{1}_{(x<y)}$ and $d^-(x,y) = d(x,y)\mathbb{1}_{(x>y)}$, where $\mathbb{1}$ is the indicator function. So we can write, when g_a is increasing,

$$\begin{aligned}
&\sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1)) \\
&= \mathbb{E}\left[\sum_{t=K}^{T-1} \mathbb{1}_{\hat{a}_{t+1}=a} \mathbb{1}_{N_a(t)\cdot d^+(\hat{\lambda}_{i,N_a(t)}, g_a(\mu_1)) \leq f(t)}\right] \\
&\leq \mathbb{E}\left[\sum_{t=K}^{T-1} \sum_{s=1}^{t} \mathbb{1}_{\hat{a}_{t+1}=a} \mathbb{1}_{N_a(t)=s} \mathbb{1}_{s\cdot d^+(\hat{\lambda}_{a,s}, g_a(\mu_1)) \leq f(T)}\right] \\
&= \mathbb{E}\Big[\sum_{s=1}^{T-1} \mathbb{1}_{s\cdot d^+(\hat{\lambda}_{a,s}, g_a(\mu_1)) \leq f(T)} \underbrace{\sum_{s=1}^{T-1} \mathbb{1}_{\hat{a}_{t+1}=a} \mathbb{1}_{N_a(t)=s}}_{\leq 1}\Big].
\end{aligned}$$

One obtains, when g_a is increasing,

$$\sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, u_a(t) \geq g_a(\mu_1)) \leq \sum_{s=1}^{T-1} \mathbb{P}\left(s \cdot d^+(\hat{\lambda}_{a,s}, g_a(\mu_1)) \leq f(T)\right). \tag{D.8}$$

Using similar arguments, one can show that when g_a is decreasing,

$$\sum_{t=K}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \ell_a(t) \leq g_a(\mu_1)) \leq \sum_{s=1}^{T-1} \mathbb{P}\left(s \cdot d^-(\hat{\lambda}_{a,s}, g_a(\mu_1)) \leq f(T)\right). \tag{D.9}$$

The quantity in the right-hand side of (D.8) is upper bounded in Appendix A.2. of Cappé et al. [2013] by

$$\frac{f(T)}{d(\lambda_a, g_a(\mu_1)} + \sqrt{2\pi}\sqrt{\frac{(d'(\lambda_a, g_a(\mu_1)))^2}{(d(\lambda_a, g_a(\mu_1)))^3}}\sqrt{f(T)} + 2\left(\frac{d'(\lambda_a, g_a(\mu_1))}{d(\lambda_a, g_a(\mu_1))}\right)^2 + 1. \tag{D.10}$$

For the second term, noting that $d^-(x,y) = d^+(1-x,1-y)$, one has

$$\begin{aligned}\mathbb{P}\left(s \cdot d^-(\hat{\lambda}_{a,s}, g_a(\mu_1)) \leq f(T)\right) &= \mathbb{P}\left(s \cdot d^+(1-\hat{\lambda}_{a,s}, 1-g_a(\mu_1)) \leq f(T)\right) \\ &= \mathbb{P}\left(s \cdot d^+(\hat{\mu}_{a,s}, 1-g_a(\mu_1)) \leq f(T)\right),\end{aligned}$$

where $\hat{\mu}_{a,s} \coloneqq 1-\hat{\lambda}_{a,s}$, is the empirical mean of s observations of a Bernoulli random variable with mean $1-\lambda_a < 1-g_a(\mu_1)$. Hence, the analysis of Cappé et al. [2013] can be applied, and using that $d(1-\lambda_a, 1-g_a(\mu_1)) = d(\lambda_a, g_a(\mu_1))$ and $d'(1-\lambda_a, 1-g_a(\mu_1)) = -d'(\lambda_a, g_a(\mu_1))$, the left hand side of (D.9) can also be upper bound by (D.10).

Combining inequalities (D.6), (D.7) and (D.8),(D.9), (D.10) with the initial decomposition of $\mathbb{E}[N_a(T)]$ yield in all cases

$$\begin{aligned}\mathbb{E}[N_a(T)] \leq\ & \frac{\log(T)}{d(\lambda_a, g_a(\mu_1))} + \sqrt{2\pi}\sqrt{\frac{d'(\lambda_a, g_a(\mu_1))^2}{d(\lambda_a, g_a(\mu_1))^3}}\sqrt{\log(T) + 3\log\log(T)} \\ &+ \left(4e + \frac{3}{d(\lambda_a, g_a(\mu_1)}\right)\log\log(T) + 2\left(\frac{d'(\lambda_a, g_a(\mu_1)}{d(\lambda_a, g_a(\mu_1)}\right)^2 + 4.\end{aligned}$$

Hence the regret of kl-UCB-CF is upper bounded by

$$\sum_{a=2}^{K} \Delta_a \Big[\frac{\log(T)}{D_a} + \sqrt{2\pi}\sqrt{\frac{(D'_a)^2}{D_a^3}}\sqrt{\log(T) + 3\log\log(T)} + \left(4e + \frac{3}{D_a}\right)\log\log(T) + 2\left(\frac{D'_a}{D_a}\right)^2 + 4\Big]$$

where $D_a \coloneqq d(\lambda_a, g_a(\mu_1))$ and $D'_a \coloneqq d'(\lambda_a, g_a(\mu_1))$, which concludes the proof.

Appendix E

Proof for Theorem 9.4

Assume 1 to be the optimal arm. For each arm non-optimal arm a, choose two thresholds u_a and w_a such that $\lambda_a < u_a < w_a < g_a(\mu_1)$ if g_a is increasing and $\lambda_a > u_a > w_a > g_a(\mu_1)$ if g_a is decreasing. Define $E_a^\lambda(t)$ as the event $\{g_a^{-1}(\hat{\lambda}_a(t)) \leq g_a^{-1}(u_a)\}$ and $E_a^\theta(t)$ as the event $\{g_a^{-1}(\theta_a(t)) \leq g_a^{-1}(w_a)\}$. Define $\mathcal{F}_t$ as the history of arm selections and received feedbacks including time t and recall that TS-CF selects the arm as follows,

$$\hat{a}_{t+1} = \operatorname*{argmax}_a \ \theta_a(t)$$

, where $\theta_a(t)$ is a sample from the posterior distribution on arm a after t observations. Define $p_{a,t} := \mathbb{P}(g_1^{-1}(\theta_1(t)) > g_a^{-1}(w_a) \mid \mathcal{F}_t)$.

We start from the following decomposition.

$$\begin{aligned}\mathbb{E}[N_a(T)] = &\sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, E_a^\lambda(t), E_a^\theta(t)) + \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, E_a^\lambda(t), \overline{E_a^\theta(t)}) \\ &+ \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \overline{E_a^\lambda(t)})\end{aligned}$$

Below are the lemmas that permit us to bound these three terms. These results generalize to the corrupted setting the main steps of the analysis of Thompson Sampling by Agrawal and Goyal [2013]. Their proof borrows a lot from that of the corresponding lemmas in this paper, with some technicalities that arise from the fact that g_1 and g_a may be either increasing or decreasing.

Lemma 9.1. $\mathbb{P}(\hat{a}_{t+1} = a, E_a^\theta(t), E_a^\lambda(t) \mid \mathcal{F}_t) \leq \frac{(1-p_{a,t})}{p_{a,t}} \mathbb{P}(\hat{a}_{t+1} = 1, E_a^\theta(t), E_a^\lambda(t) \mid \mathcal{F}_t)$.

Proof. Assume that $E_a^\lambda(t)$ is true (otherwise the lemma holds trivially because the left hand size is 0). Hence, it is sufficient to prove that,

$$\mathbb{P}(\hat{a}_{t+1} = a \mid E_a^\theta(t), \mathcal{F}_t) \leq \frac{(1 - p_{a,t})}{p_{a,t}} \mathbb{P}(\hat{a}_{t+1} = 1 \mid E_a^\theta(t), \mathcal{F}_t))$$

Define $M_a(t)$ the event in which the index of arm a at time t is the largest among those of all suboptimal arms: $M_a(t) := \Big\{ g_a^{-1}(\theta_a(t)) \geq g_j^{-1}(\theta_j(t)), \forall j \neq 1 \Big\}$.

$$\begin{aligned}
&\mathbb{P}(\hat{a}_{t+1} = 1 \mid E_a^\theta(t), \mathcal{F}_t)) \\
&\geq \mathbb{P}(\hat{a}_{t+1} = 1, M_a(t) \mid E_a^\theta(t), \mathcal{F}_t)) \\
&= \mathbb{P}(M_a(t) \mid E_a^\theta(t), \mathcal{F}_t)) \cdot \mathbb{P}(\hat{a}_{t+1} = 1 \mid M_a(t), E_a^\theta(t), \mathcal{F}_t))
\end{aligned} \tag{E.1}$$

Now, given $M_a(t)$ and $E_a^\theta(t)$ hold,

$$g_j^{-1}(\theta_j(t)) \leq g_a^{-1}(\theta_a(t)) \leq g_a^{-1}(w_a) \quad \forall j \neq a, j \neq 1$$

So,

$$\begin{aligned}
\mathbb{P}(\hat{a}_{t+1} = 1 \mid M_a(t), E_a^\theta(t), \mathcal{F}_t) &\geq \mathbb{P}(g_1^{-1}(\theta_1(t)) > g_a^{-1}(w_a) \mid M_a(t), E_a^\theta(t), \mathcal{F}_t) \\
&= \mathbb{P}(g_1^{-1}(\theta_1(t)) > g_a^{-1}(w_a) \mid \mathcal{F}_t) \\
&= p_{a,t}
\end{aligned} \tag{E.2}$$

From inequalities (E.1) and (E.2),

$$\mathbb{P}(\hat{a}_{t+1} = 1 \mid E_a^\theta(t), \mathcal{F}_t) \geq p_{a,t} \cdot \mathbb{P}(M_a(t) \mid E_a^\theta(t), \mathcal{F}_t) \tag{E.3}$$

Now, let us consider the left hand side of the inequality. The fact that $E_a^\theta(t)$ holds and $\hat{a}_{t+1} = a$ implies that $g_1^{-1}(\theta_1(t)) < g_a^{-1}(\theta_a(t)) < g_a^{-1}(w_a)$. Hence

$$\begin{aligned}
&\mathbb{P}(\hat{a}_{t+1} = a \mid E_a^\theta(t), \mathcal{F}_t) \\
&\leq \mathbb{P}\Big(g_1^{-1}(\theta_1(t)) \leq g_a^{-1}(w_a), g_a^{-1}(\theta_a(t)) \geq g_j^{-1}(\theta_j(t)), \forall j \neq 1 \mid E_a^\theta(t), \mathcal{F}_t\Big) \\
&= \mathbb{P}\Big(g_1^{-1}(\theta_1(t)) \leq g_a^{-1}(w_a) \mid \mathcal{F}_{t-1}\Big) \cdot \mathbb{P}\Big(g_a^{-1}(\theta_a(t)) \geq g_j^{-1}(\theta_j(t)), \forall j \neq 1 \mid E_a^\theta(t), \mathcal{F}_t\Big)
\end{aligned}$$

$$= (1 - p_{a,t}) \cdot \mathbb{P}(M_a(t) \mid E_a^\theta(t), \mathcal{F}_t) \tag{E.4}$$

From inequalities (E.3) and (E.4),

$$\mathbb{P}(\hat{a}_{t+1} = a \mid E_a^\theta(t), \mathcal{F}_t) \leq \frac{(1 - p_{a,t})}{p_{a,t}} \mathbb{P}(\hat{a}_{t+1} = 1 \mid E_a^\theta(t), \mathcal{F}_t)$$

□

Lemma 9.2. *When g_a is increasing (resp. decreasing), for any $u'_a \in (u_a, w_a)$ (resp. (w_a, u_a)), when T is large enough,*

$$\sum_{t=0}^{T-1} \mathbb{P}\Big(\hat{a}_{t+1} = a, \overline{E_a^\theta(t)}, E_a^\lambda(t)\Big) \leq \frac{\log(T)}{d(u'_a, w_a)} + 1.$$

Proof. When g_a is increasing, the application of Lemma 3 in Agrawal and Goyal [2013] directly yields

$$\sum_{t=0}^{T-1} \mathbb{P}\left(\hat{a}_{t+1} = a, \overline{E_a^\theta(t)}, E_a^\lambda(t)\right) \leq \frac{\log T}{d(u_a, w_a)} + 1.$$

The proof is based on the use of deviation inequalities and a link between the Beta and Binomial c.d.f. that shall also be useful in the decreasing case, that we handle now (using slightly different arguments).

Fact E.1.

$$F_{\alpha,\beta}^{beta}(w) = 1 - F_{\alpha+\beta-1,w}^{B}(\alpha - 1)$$

Note that for decreasing g_a, one has $\overline{E_a^\theta(t)} = \{\theta_a(t) \leq w_a\}$ and $E_a^\lambda(t) = \{\hat{\lambda}_a(t) > u_a\}$. Fix u'_a such that $w_a < u'_a < u_a$ and let $L'_a(T) = \frac{\log(T)}{d(u'_a, w_a)}$.

$$\begin{aligned}
&\sum_{t=0}^{T-1} \mathbb{P}\left(\hat{a}_{t+1} = a, \hat{\lambda}_a(t) > u_a, \theta_a(t) \leq w_a\right) \\
&\leq \frac{\log(T)}{d(u'_a, w_a)} + \sum_{t=0}^{T-1} \mathbb{P}\left(\hat{a}_{t+1} = a, N_a(t) \leq L'_a(T), \theta_a(t) \leq w_a, \hat{\lambda}_a(t) > u_a\right) \\
&\leq \frac{\log(T)}{d(u'_a, w_a)} + \mathbb{E} \sum_{t=0}^{T-1} \sum_{s=L'_a(T)}^{t} \mathbb{1}_{(\hat{a}_{t+1}=a, N_a(t)=s, \theta_a(t)\leq w_a, \hat{\lambda}_a(t) > u_a)}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\log(T)}{d(u'_a, w_a)} + \mathbb{E}\sum_{t=0}^{T-1}\sum_{s=L'_a(T)}^{t} \mathbb{1}_{\left(\hat{a}_{t+1}=a, N_a(t)=s, \hat{\lambda}_a(t)>u_a\right)} \mathbb{P}\left(\theta_a(t) \leq w_a \mid \mathcal{F}_t\right) \\
&= \frac{\log(T)}{d(u'_a, w_a)} + \mathbb{E}\sum_{t=0}^{T-1}\sum_{s=L'_a(T)}^{t} \mathbb{1}_{\left(\hat{a}_{t+1}=a, N_a(t)=s, \hat{\lambda}_a(t)>u_a\right)} F^{beta}_{(s\hat{\lambda}_a(t)+1, s-s\hat{\lambda}_a(t)+1)}(w_a) \\
&= \frac{\log(T)}{d(u'_a, w_a)} + \mathbb{E}\sum_{t=0}^{T-1}\sum_{s=L'_a(T)}^{t} \mathbb{1}_{\left(\hat{a}_{t+1}=a, N_a(t)=s, \hat{\lambda}_a(t)>u_a\right)} \left(1 - F^{B}_{(s+1, w_a)}(s\hat{\lambda}_a(t))\right) \\
&\leq \frac{\log(T)}{d(u'_a, w_a)} + \mathbb{E}\sum_{t=0}^{T-1}\sum_{s=L'_a(T)}^{t} \mathbb{1}_{\left(\hat{a}_{t+1}=a, N_a(t)=s, \hat{\lambda}_a(t)>u_a\right)} \underbrace{\left(1 - F^{B}_{(s+1, w_a)}(su_a)\right)}_{A_s}
\end{aligned}
$$

Introducing (X_k) an i.i.d. sequence drawn from Bernoulli of mean w_a, term A_s can be written, for any s, .

$$
\begin{aligned}
A_s &= \mathbb{P}\left(\sum_{k=1}^{s+1} X_k \geq u_a s\right) \leq \mathbb{P}\left(\sum_{k=1}^{s} X_k \geq u_a s - 1\right) = \mathbb{P}\left(\frac{1}{s}\sum_{k=1}^{s} X_k \geq u_a - \frac{1}{s}\right) \\
&\leq \exp\left(-sd\left(u_a - 1/s, w_a\right)\right) \leq \exp\left(-\log(T)\frac{d(u_a - 1/s, w_a)}{d(u'_a, w_a)}\right) \leq \frac{1}{T},
\end{aligned}
$$

for large enough T, and s larger than $L'_a(T)$ (as it holds that $d(u_a - 1/s, w_a) \geq d(u'_a, w_a)$). Finally, for T large enough,

$$
\begin{aligned}
&\sum_{t=1}^{T} \mathbb{P}\left(\hat{a}_t = a, \hat{\lambda}_a(t) \geq u_a, \theta_a(t) \leq w_a\right) \\
&\leq \frac{\log(T)}{d(u'_a, w_a)} + \sum_{s=0}^{T-1} \frac{1}{T} \mathbb{E} \underbrace{\sum_{t=s}^{T} \mathbb{1}_{(\hat{a}_{t+1}=a, N_a(t)=s)}}_{\leq 1} \\
&\leq \frac{\log(T)}{d(u'_a, w_a)} + \sum_{t=1}^{T} \frac{1}{T} = \frac{\log(T)}{d(u'_a, w_a)} + 1.
\end{aligned}
$$

□

Lemma 9.3. $\sum_{t=0}^{T-1} \mathbb{P}\left(\hat{a}_{t+1} = a, \overline{E_a^{\lambda}(t)}\right) \leq 1 + \frac{1}{d(u_a, \lambda_a)}$.

Proof. This result follows from the application of Chernoff bound for the concentration of $\hat{\lambda}_a(t)$. When g_a is increasing, it follows directly from the application of Lemma 2 in Agrawal and Goyal [2013], hence we write the proof in the decreasing

case only, were we shall justify that for $u_a < \lambda_a$,

$$\sum_{t=0}^{T-1} \mathbb{P}\left(\hat{a}_{t+1}, \hat{\lambda}_a(t) < u_a\right) \leq \frac{1}{d(u_a, \lambda_a)} + 1.$$

Using $\hat{\lambda}_{a,s}$ to denote the empirical mean of the s first observations from the feedback of arm a,

$$\begin{aligned}
\sum_{t=0}^{T-1} \mathbb{P}\left(\hat{a}_{t+1}, \hat{\lambda}_a(t) < u_a\right) &= \mathbb{E}\left[\sum_{t=0}^{T-1}\sum_{s=0}^{t} \mathbb{1}_{(\hat{a}_{t+1}=a, N_a(t)=s)} \mathbb{1}_{(\hat{\lambda}_{a,s}<u_a)}\right] \\
&= \mathbb{E}\Big[\sum_{s=0}^{T} \mathbb{1}_{\left(\hat{\lambda}_{a,s}<u_a\right)} \underbrace{\sum_{t=s}^{T} \mathbb{1}_{(\hat{a}_{t+1}=a, N_a(t)=s)}}_{\leq 1}\Big] \\
&\leq 1 + \sum_{s=1}^{T-1} \mathbb{P}\left(\hat{\lambda}_{a,s} < u_a\right) \leq 1 + \sum_{s=1}^{T-1} \exp(-s d(u_a, \lambda_a)) \\
&\leq 1 + \frac{1}{d(u_a, \lambda_a)},
\end{aligned}$$

where the last but one inequality follows from Chernoff inequality (as $u_a < \lambda_a$). □

Lemma 9.4. *Let τ_s be the instant of the s-th play of arm 1. Then there exists a function $f(s) \coloneqq f(s, \lambda_1, g_1(g_a^{-1}(\mu_1)))$ satisfying $\sum_{s=1}^{\infty} f(s) < \infty$ such that for all s,*

$$\mathbb{E}\left[\frac{1}{p_{a,\tau_s+1}}\right] \leq 1 + f(s).$$

Proof. Let $\tilde{w}_a \coloneqq g_1(g_a^{-1}(w_a))$. Examining all possibilities, one can easily show that

- if g_1 is increasing and g_a is increasing, $p_{a,t} = \mathbb{P}\left(\theta_1(t) > \tilde{w}_a\right)$, with $\tilde{w}_a < \lambda_1$,
- if g_1 is increasing and g_a decreasing, $p_{a,t} = \mathbb{P}\left(\theta_1(t) > \tilde{w}_a\right)$, with $\tilde{w}_a < \lambda_1$,
- if g_1 is decreasing and g_a is increasing, $p_{a,t} = \mathbb{P}\left(\theta_1(t) < \tilde{w}_a\right)$, with $\tilde{w}_a > \lambda_1$,
- if g_1 is decreasing and g_a is decreasing, $p_{a,t} = \mathbb{P}\left(\theta_1(t) < \tilde{w}_a\right)$, with $\tilde{w}_a > \lambda_1$.

When g_1 is increasing, $\tilde{w}_a < \lambda_1$ and

$$p_{a,\tau_s+1} = 1 - F^{beta}_{(S_1(\tau_s)+1, s-S_1(\tau_s)+1)}(\tilde{w}_a) = F^{B}_{(s+1,\tilde{w}_a)}(S_1(\tau_s)).$$

Using that $S_1(\tau_s)$ has a binomial distribution with parameters (s, λ_1) yields

$$\mathbb{E}\left[\frac{1}{p_{a,\tau_s+1}}\right] = \sum_{j=0}^{s} \frac{f^B_{(s,\lambda_1)}(j)}{F^B_{(s+1,\tilde{w}_a)}(j)}. \tag{E.5}$$

When g_1 is decreasing, recall $\tilde{w}_a > \lambda_1$ and one has

$$p_{a,\tau_s+1} = F^{beta}_{(S_1(\tau_s)+1,s-S_1(\tau_s)+1)}(\tilde{w}_a) = 1 - F^B_{(s+1,\tilde{w}_a)}(S_1(\tau_s)).$$

Using again the distribution of $S_1(\tau_s)$ yields

$$\mathbb{E}\left[\frac{1}{p_{a,\tau_s+1}}\right] = \sum_{j=0}^{s} \frac{f^B_{(s,\lambda_1)}(j)}{1 - F^B_{(s+1,\tilde{w}_a)}(j)}$$

Note here two simple properties of Binomial distributions: for all $t \in \mathbb{N}^*$ and $c \in [0,1]$, for all $j \in \{0, \dots, t\}$,

- $f^B_{(t,c)}(j) = f_{(t,1-c)}(s-j)$
- $F^B_{(t,c)}(j) = 1 - F^B_{(t,1-c)}(t-j-1)$

It follows that

$$\mathbb{E}\left[\frac{1}{p_{a,\tau_s+1}}\right] = \sum_{j=0}^{s} \frac{f^B_{(s,1-\lambda_1)}(s-j)}{F^B_{(s+1,1-\tilde{w}_a)}(s-j)} = \sum_{j=0}^{s} \frac{f^B_{(s,1-\lambda_1)}(j)}{F^B_{(s+1,1-\tilde{w}_a)}(j)}, \tag{E.6}$$

with $1 - \lambda_1 > 1 - \tilde{w}_a$.

The proof for Lemma 4 given in Agrawal and Goyal [2013] provides an upper bound on the quantity

$$\sum_{j=0}^{s} \frac{f^B_{(s,c)}(j)}{F^B_{(s+1,c)}(j)}$$

whenever c is larger that d. Using this result one can bound (E.5) and (E.6) by the same quantity:

$$\mathbb{E}\left[\frac{1}{p_{a,\tau_s+1}}\right] \leq \begin{cases} 1 + \frac{3}{\Delta'_a}, & \text{if } s < \frac{8}{\Delta'_a} \\ 1 + \Theta\left(\exp\left(-{\Delta'_a}^2 s/2\right) + \frac{1}{(s+1){\Delta'_a}^2}\exp\left(-D_a s\right) + \frac{1}{\exp\left({\Delta'_a}^2 s/4\right)-1}\right), & \text{if } s \geq \frac{8}{\Delta'_a} \end{cases}$$

where $\Delta'_a \coloneqq \lambda_1 - \tilde{w}_a$ and $D_a \coloneqq \tilde{w}_a \log \frac{\tilde{w}_a}{\lambda_1} + (1-\tilde{w}_a)\log\frac{1-\tilde{w}_a}{1-\lambda_1}$. Hence, Lemma 9.4 follows with

$$f(s) \coloneqq \begin{cases} \frac{3}{\Delta'_a}, & \text{if } s < \frac{8}{\Delta'_a} \\ \Theta\left(\exp\left(-{\Delta'_a}^2 s/2\right) + \frac{1}{(s+1){\Delta'_a}^2}\exp\left(-D_a s\right) + \frac{1}{\exp\left({\Delta'_a}^2 s/4\right)-1}\right), & \text{if } s \geq \frac{8}{\Delta'_a} \end{cases},$$

that satisfies $\sum_{s=0}^{\infty} f(s) < \infty$. □

One can now complete the proof of Theorem 9.4.

$$\begin{aligned}
\mathbb{E}[N_a(T)] &= \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a) \\
&= \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, E_a^{\lambda}(t), E_a^{\theta}(t)) + \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, E_a^{\lambda}(t), \overline{E_a^{\theta}(t)}) \\
&+ \sum_{t=0}^{T-1} \mathbb{P}(\hat{a}_{t+1} = a, \overline{E_a^{\lambda}(t)}) \\
&\leq \sum_{t=0}^{T-1} \mathbb{E}\left[\frac{(1-p_{a,t})}{p_{a,t}} \mathbb{1}_{(\hat{a}_{t+1}=1, E_a^{\theta}(t), E_a^{\lambda}(t))}\right] + \frac{\log T}{d(u'_a, w_a)} + 1 + \frac{1}{d(u_a,\lambda_a)} + 1 \\
&\leq \sum_{s=0}^{T-1} \mathbb{E}\left[\frac{(1-p_{a,\tau_s+1})}{p_{a,\tau_s+1}} \sum_{t=\tau_s}^{\tau_{s+1}-1} \mathbb{1}_{(\hat{a}_{t+1}=1)}\right] + \frac{\log T}{d(u'_a, w_a)} + 1 + \frac{1}{d(u_a,\lambda_a)} + 1 \\
&= \sum_{s=0}^{T-1} \mathbb{E}\left[\frac{1}{p_{a,\tau_s+1}} - 1\right] + \frac{\log T}{d(u'_a, w_a)} + 1 + \frac{1}{d(u_a,\lambda_a)} + 1 \\
&\leq \frac{\log T}{d(u'_a, w_a)} + \sum_{s=0}^{T-1} f(s) + \frac{1}{d(u_a,\lambda_a)} + 2.
\end{aligned}$$

Fix $\epsilon > 0$. Using the monotonicity properties of the divergence function d, there exists $u_a < u'_a < w_a$ in the increasing case and $u_a > u'_a > w_a$ in the decreasing case such that $d(u'_a, w_a) \geq d(\lambda_a, g_a(\mu_1))/(1+\epsilon)$. For these particular choice, one obtains

$$\mathbb{E}[N_a(T)] \leq (1+\epsilon)\frac{\log(T)}{d(\lambda_a, g_a(\mu_a))} + R(u_a, u'_a, w_a),$$

where $R(u_a, u'_a, w_a)$ is a rest term that depends on $\epsilon, \mu_1, \mu_a, g_1$ and g_a. The result follows using that $\mathrm{CRegret}_T = \sum_{a=2}^{K} \Delta_a \mathbb{E}[N_a(T)]$.

Part IV

Final Remarks

Chapter 13

Summary and future work

In Part I of this thesis, we briefly introduce the sequential decision making problem, of which the MAB problem is a special case. We explain how the relaxation of the assumption of complete feedback to bandit feedback in sequential decision making leads us a MAB problem. We formally define the MAB problem along with its various formalizations depending upon how the reward is generated. We also specify various goals for the learner and characterize them as performance measures e.g. regret and sample complexity. We briefly outline the practical applications which can be modeled as a MAB problem due to the availability of the conventional bandit feedback. We also study some of the classical algorithms for the MAB problem which form the basis of the algorithms we introduce later in the thesis. Additionally, we illustrate a general paradigm of partial monitoring games which can be used for describing MAB problems with unconventional feedback introduced in this thesis.

In Part II of this thesis, we concentrate on the first kind of unconventional feedback considered : relative feedback. We motivate the necessity of using relative feedback by providing practical applications where it is used. We then formally define the MAB problem with relative feedback or the dueling bandit problem. We also illustrate how the dueling bandit problem can be formalized in each of the various settings described in the first part. We consider a problem setting rarely studied in the previous related work: adversarial utility-based dueling bandits. We provide the lower bound on the regret of any algorithm for this setting. We propose a novel algorithm called, Relative Exponential-weight algorithm for Exploration and Exploitation (REX3) and prove the upper bound on its regret. We compare the performance of REX3 with the state of the art algorithms using the experiments performed on real

datasets and on simulations too. These experiments show that REX3is a suitable solution for the adversarial utility-based dueling bandits. In the end, we formulate the dueling bandits problem as an instance of the partial monitoring game. Furthermore, we show that the existing partial monitoring algorithms are suboptimal in terms of the the number of arms.

In Part III of this thesis, we study another kind of unconventional feedback: corrupted feedback. We explain how the motivation from corrupted feedback arises from practical applications. We formally defined the MAB problem with corrupted feedback or the corrupt bandits problem. We provide the lower bounds on the regret for exploration-exploitation setting and on the sample complexity for best arm identification. We propose two algorithms for best arm identification: Median elimination for corrupt bandits (ME-CF) and Exponential-gap elimination for corrupt bandits (EGE-CF). We prove the upper bounds on their sample complexity. We also propose two algorithms for exploration-exploitation: kl-UCB-CF and TS-CF and prove the upper bounds on their regret. Comparing the upper bound on regret with the lower bound shows that both kl-UCB-CFand TS-CF and asymptotically optimal. We demonstrate how corrupted feedback can be used for privacy preservation at the user level in online recommender systems. We also provide the appropriate corruption parameters to guarantee a desired level of differential privacy and analyze how this impacts the regret. Finally, we present experimental results which confirm our theoretical results.

The work on these problems paves the way to study other kinds of unconventional feedback inspired from practical applications and to solve a number of closely-connected problems. In the two kinds of feedback considered in this thesis as well, immediate extensions are possible. For dueling bandits, a natural extension is to study adversarial preference-based formulation. In the setting of corrupted feedback, the feedback is available at all times. In some situations however, the feedback is simply lost. It is possible extend our problem setting to incorporate such scenarios by making appropriate changes to the corruption process. An adversarial corruption of the feedback can be considered too. Another possible extension is to incorporate contextual information in the learning process with preference feedback or corrupted feedback. Since both the kinds of feedback can be expressed as a partial

monitoring game, it is worthwhile to see if it is possible to devise a general partial monitoring algorithm which could deal with such unconventional feedback. We illustrated in Chapter 6 that the bounds on regret for the current partial monitoring algorithms are not tight enough in terms of the number of arms. It remains an open problem to see if a general partial monitoring algorithm could achieve the optimality in terms of the number of arms.

Bibliography

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 263–274, 2008. URL http://colt2008.cs.helsinki.fi/papers/127-Abernethy.pdf.

Alekh Agarwal, Peter L. Bartlett, and Max Dama. Optimal allocation strategies for the dark pool problem. In Yee Whye Teh and D. Mike Titterington, editors, *AISTATS*, volume 9 of *JMLR Proceedings*, pages 9–16. JMLR.org, 2010. URL http://dblp.uni-trier.de/db/journals/jmlr/jmlrp9.html#AgarwalBD10.

Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th annual international ACM conference on Research and development in information retrieval (SIGIR)*, pages 3–10. ACM, 2006.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. *Journal of Machine Learning Research - Proceedings Track*, 23: 39.1–39.26, 2012.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, pages 99–107, 2013. URL http://jmlr.org/proceedings/papers/v31/agrawal13a.html.

N. Ailon, Z. Shay Karnin, and T. Joachims. Reducing dueling bandits to cardinal bandits. In *ICML 2014*, volume 32 of *JMLR Proceedings*, pages 856–864, 2014.

Dana Angluin and Philip Laird. Learning from noisy examples. *Mach. Learn.*, 2 (4):343–370, April 1988. ISSN 0885-6125. doi: 10.1023/A:1022873112823. URL http://dx.doi.org/10.1023/A:1022873112823.

J.Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *COLT*, Haïfa (Israël), 2010. Omnipress.

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

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

Naveen Farag Awad and M. S. Krishnan. The personalization privacy paradox: An empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS Q.*, 30(1):13–28, March 2006. ISSN 0276-7783. URL http://dl.acm.org/citation.cfm?id=2017284.2017287.

Gábor Bartók. *The Role of Information in Online Learning*. PhD thesis, Edmonton, Alta., Canada, 2012. AAINR89904.

Gábor Bartók. A near-optimal algorithm for finite partial-monitoring games against adversarial opponents. In *Proc. COLT*, 2013.

Gábor Bartók, Dvid Pál, and Csaba Szepesvári. Minimax regret of finite partial-monitoring games in stochastic environments. In *In Conference on Learning Theory*, 2011.

Gábor Bartók, Dean P. Foster, Dávid Pál, Alexander Rakhlin, and Csaba Szepesvári. Partial monitoring - classification, regret bounds, and algorithms. *Math. Oper. Res.*, 39(4):967–997, 2014.

Robert E. Bechhofer. A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs. *Biometrics*, 14(3):408–429, 1958. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2527883.

J. Bennett and S. Lanning. The netflix prize. In *Proceedings of the KDD Cup Workshop 2007*, pages 3–6, New York, August 2007. ACM. URL http://www.cs.uic.edu/~liub/KDD-cup-2007/NetflixPrize-description.pdf.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. doi: 10.1561/2200000024. URL http://dx.doi.org/10.1561/2200000024.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *ALT*, pages 23–37, 2009. doi: 10.1007/978-3-642-04414-4_7.

Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. "you might also like: " privacy risks of collaborative filtering. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*, pages 231–246, 2011. doi: 10.1109/SP.2011.40. URL http://dx.doi.org/10.1109/SP.2011.40.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar F. Zaidan. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 22–64, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-12-1. URL http://dl.acm.org/citation.cfm?id=2132960.2132964.

O. Cappé, A. Garivier, O-A. Maillard, R. Munos, and G. Stoltz. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics*, 41(3):1516–1541, 2013.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089.

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Mach. Learn.*, 66(2-3):321–352, March 2007.

Nicolò Cesa-bianchi, Gábor Lugosi, and Gilles Stoltz. Minimizing regret with label efficient prediction. *IEEE Trans. Inform. Theory*, 51:77–92, 2005.

T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26:1–26:24, November 2011. ISSN 1094-9224. doi: 10.1145/2043621.2043626. URL http://doi.acm.org/10.1145/2043621.2043626.

O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Trans. Inf. Syst.*, 30(1):6, 2012.

I. Charon and O. Hudry. An updated survey on the linear ordering problem for weighted or unweighted tournaments. *Annals OR*, 175(1):107–158, 2010. doi: 10.1007/s10479-009-0648-7.

Ramnath K. Chellappa and Raymond G. Sin. Personalization versus privacy: An empirical examination of the online consumer's dilemma. *Inf. Technol. and Management*, 6(2-3):181–202, April 2005. ISSN 1385-951X. doi: 10.1007/s10799-005-5879-y. URL http://dx.doi.org/10.1007/s10799-005-5879-y.

V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. *Microdata Protection*, pages 291–321. Springer US, Boston, MA, 2007. ISBN 978-0-387-27696-0. doi: 10.1007/978-0-387-27696-0_9. URL http://dx.doi.org/10.1007/978-0-387-27696-0_9.

Mary J. Culnan. Protecting Privacy Online: Is Self-Regulation Working? *Journal of Public Policy & Marketing*, 19(1):20–26, 2000.

Ofer Dekel, Ohad Shamir, and Lin Xiao. Learning to classify with missing and corrupted features. *Machine Learning*, 81(2):149–178, 2010. doi: 10.1007/s10994-009-5124-8. URL http://dx.doi.org/10.1007/s10994-009-5124-8.

FranÇois Denis. *Algorithmic Learning Theory: 9th International Conference, ALT'98 Otzenhausen, Germany, October 8–10, 1998 Proceedings*, chapter PAC Learning from Positive Statistical Queries, pages 112–126. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-49730-1. doi: 10.1007/3-540-49730-7_9. URL http://dx.doi.org/10.1007/3-540-49730-7_9.

François Denis, Rémi Gilleron, and Marc Tommasi. Text Classification from Positive and Unlabeled Examples. In *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'02*, pages 1927–1934, Grenoble, France, 2002. URL https://hal.inria.fr/inria-00538889.

Francois Denis, Anne Laurent, Rémi Gilleron, and Marc Tommasi. Text classification and co-training from positive and unlabeled examples. In *Proceedings of the ICML 2003 Workshop: The Continuum from Labeled to Unlabeled Data*, pages 80–87, 2003.

François Denis, Rémi Gilleron, and Fabien Letouzey. Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348(1):70 – 83, 2005. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/j.tcs.2005.09.007. URL http://www.sciencedirect.com/science/article/pii/S0304397505005256. Algorithmic Learning Theory (ALT 2000)11th International Conference, Algorithmic Learning Theory 2000.

Miroslav Dudík, Katja Hofmann, Robert E. Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In Peter Grünwald, Elad Hazan,

and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 563–587, Paris, France, 03–06 Jul 2015. PMLR. URL http://proceedings.mlr.press/v40/Dudik15.html.

Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052, pages 1–12, Venice, Italy, July 2006. Springer Verlag. ISBN 3-540-35907-9. URL https://www.microsoft.com/en-us/research/publication/differential-privacy/.

Cynthia Dwork. The Differential Privacy Frontier (Extended Abstract) Theory of Cryptography. In Omer Reingold, editor, *Theory of Cryptography*, volume 5444 of *Lecture Notes in Computer Science*, chapter 29, pages 496–502. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-00456-8. doi: 10.1007/978-3-642-00457-5_29. URL http://dx.doi.org/10.1007/978-3-642-00457-5_29.

Cynthia Dwork. Differential privacy in new settings. In *Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, January 2010. URL https://www.microsoft.com/en-us/research/publication/differential-privacy-in-new-settings/.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9:211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL http://dx.doi.org/10.1561/0400000042.

Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *In Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 381–390, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536467. URL http://doi.acm.org/10.1145/1536414.1536467.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

Uriel Feige, Yishay Mansour, and Robert E. Schapire. Learning and inference in the presence of corrupted inputs. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 637–657, 2015. URL http://jmlr.org/proceedings/papers/v40/Feige15.html.

Abraham Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *CoRR*, cs.LG/0408007, 2004. URL http://arxiv.org/abs/cs.LG/0408007.

Dean P. Foster and Alexander Rakhlin. No internal regret via neighborhood watch. *CoRR*, abs/1108.6088, 2011. URL http://arxiv.org/abs/1108.6088.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Netw. Learning Syst.*, 25(5):845–869, 2014. doi: 10.1109/TNNLS.2013.2292894. URL http://dx.doi.org/10.1109/TNNLS.2013.2292894.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997. ISSN 0022-0000. doi: 10.1006/jcss.1997.1504.

Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999a. URL http://EconPapers.repec.org/RePEc:eee:gamebe:v:29:y:1999:i:1-2:p:79-103.

Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1):79–103, October 1999b. URL http://www.cs.princeton.edu/~{}schapire/publist.html.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, December 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=945365.964285.

Johannes Fürnkranz and Eyke Hüllermeier, editors. *Preference Learning*. Springer-Verlag, 2010. ISBN 978-3642141249. doi: 10.1007/978-3-642-14125-6. URL http://www.springer.com/978-3-642-14124-9.

Pratik Gajane, Tanguy Urvoy, and Fabrice Clérot. A relative exponential weighing algorithm for adversarial utility-based dueling bandits. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 218–227, 2015. URL http://jmlr.org/proceedings/papers/v37/gajane15.html.

Aurélien Garivier, Gilles Stoltz, and Pierre Ménard. Explore first, exploit next: The true shape of regret in bandit problems. 2016.

Marco De Gemmis, Leo Iaquinta, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Preference learning in recommender systems. In *In Preference Learning (PL-09) ECML/PKDD-09 Workshop*, 2009.

Amir Globerson and Sam Roweis. Nightmare at test time: Robust learning by feature deletion. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 353–360, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143889. URL http://doi.acm.org/10.1145/1143844.1143889.

Jim Harper. It's modern trade: Web users get as much as they give. *The Wall Street Journal*, August 7, 2010.

Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 24.1–24.34, 2012.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2), April 2007.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, October 2005. ISSN 0022-0000. doi: 10.1016/j.jcss.2004.10.016. URL http://dx.doi.org/10.1016/j.jcss.2004.10.016.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 1238–1246. JMLR Workshop and Conference Proceedings, May 2013. URL http://jmlr.org/proceedings/papers/v28/karnin13.pdf.

Richard M. Karp and Robert Kleinberg. Noisy binary search and its applications. In *SODA 2007*, SIAM Proceedings, pages 881–890, 2007. ISBN 978-0-898716-24-5. URL http://dl.acm.org/citation.cfm?id=1283383.1283478.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42, 2016. URL http://jmlr.org/papers/v17/kaufman16a.html.

Michael Kearns and Ming Li. Learning in the presence of malicious errors. *SIAM J. Comput.*, 22(4):807–837, August 1993. ISSN 0097-5397. doi: 10.1137/0222052. URL http://dx.doi.org/10.1137/0222052.

Jay Kim. A method for limiting disclosure in microdata based on random noise and transformation. In *Proceedings of the Survey Research Methods, American Statistical Association*, pages 370–374, 1986.

Jay J. Kim, Jay J. Kim, William E. Winkler, and William E. Winkler. Multiplicative noise for masking continuous data. Technical report, Statistical Research Division, US Bureau of the Census, Washington D.C, 2003.

Robert D. Kleinberg and Frank Thomson Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *FOCS*, pages 594–605. IEEE Computer Society, 2003. ISBN 0-7695-2040-5. URL http://dblp.uni-trier.de/db/conf/focs/focs2003.html#KleinbergL03.

Aleksandra Korolova. Privacy violations using microtargeted ads: A case study. In *ICDMW 2010, The 10th IEEE International Conference on Data Mining Workshops, Sydney, Australia, 13 December 2010*, pages 474–482, 2010. doi: 10.1109/ICDMW.2010.137. URL http://dx.doi.org/10.1109/ICDMW.2010.137.

Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

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

Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, pages 448–455, 2003a.

Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML*, page 2003, 2003b.

Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 587–592, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1630659.1630746.

Yehuda Lindell and Eran Omri. A practical application of differential privacy to personalized online advertising. *IACR Cryptology ePrint Archive*, 2011:152, 2011. URL http://eprint.iacr.org/2011/152.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, February 1994. ISSN 0890-5401. doi: 10.1006/inco.1994.1009. URL http://dx.doi.org/10.1006/inco.1994.1009.

Michael L Littman. Algorithms for sequential decision making. Technical report, Providence, RI, USA, 1996.

Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 387–394, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1-55860-873-7. URL http://dl.acm.org/citation.cfm?id=645531.656022.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 179–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1978-4. URL http://dl.acm.org/citation.cfm?id=951949.952139.

Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3 (3):225–331, March 2009. ISSN 1554-0669. doi: 10.1561/1500000016. URL http://dx.doi.org/10.1561/1500000016.

Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR 2007*. ACM, 2007.

Nikita Mishra and Abhradeep Thakurta. (nearly) optimal differentially private stochastic multi-arm bandits. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, pages 592–601, 2015.

Kato Mivule. Utilizing noise addition for data privacy, an overview. *CoRR*, abs/1309.3958, 2013. URL http://arxiv.org/abs/1309.3958.

MSLR30K. Microsoft learning to rank dataset, 2012. URL http://research.microsoft.com/en-us/projects/mslr/default.aspx.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1196–1204. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5073-learning-with-noisy-labels.pdf.

Edward Paulson. A sequential procedure for selecting the population with the largest mean from k normal populations. *Ann. Math. Statist.*, 35(1):174–180, 03

1964. doi: 10.1214/aoms/1177703739. URL http://dx.doi.org/10.1214/aoms/1177703739.

Antonio Piccolboni and Christian Schindelhauer. Discrete prediction games with arbitrary feedback and loss. In *COLT/EuroCOLT*, volume 2111 of *LNCS*, pages 208–223. Springer, 2001.

F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *KDD 2007*, pages 570–579. ACM, 2007. doi: 10.1145/1281192.1281254.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 1952.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951. doi: 10.1214/aoms/1177729586. URL http://dx.doi.org/10.1214/aoms/1177729586.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

José A. Sáez, Mikel Galar, Julián Luengo, and Francisco Herrera. Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. *Knowledge and Information Systems*, 38(1):179–206, 2014. ISSN 0219-3116. doi: 10.1007/s10115-012-0570-1. URL http://dx.doi.org/10.1007/s10115-012-0570-1.

Yevgeny Seldin, Csaba Szepesvári, Peter Auer, and Yasin Abbasi-Yadkori. Evaluation and analysis of the performance of the exp3 Algorithm in stochastic environments. In *EWRL*, volume 24 of *JMLR Proceedings*, pages 103–116, 2012. URL http://dblp.uni-trier.de/db/conf/ewrl/ewrl2012.html#SeldinSAA12.

Abhradeep Guha Thakurta and Adam D. Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2733–2741, 2013.

W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Bulletin of the AMS*, 25:285–294, 1933.

Aristide C. Y. Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. In *13th International Conference on Artificial Intelligence (AAAI 2016)*, 2016.

T. Urvoy, F. Clerot, R. Féraud, and S. Naamane. Generic exploration and K-armed voting bandits. In *ICML 2013*, volume 28 of *JMLR Proceedings*, pages 91–99, 2013a.

Tanguy Urvoy, Fabrice Clerot, Raphael Féraud, and Sami Naamane. Generic exploration and k-armed voting bandits. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 91–99. JMLR Workshop and Conference Proceedings, May 2013b. URL http://jmlr.csail.mit.edu/proceedings/papers/v28/urvoy13.pdf.

Abraham Wald. On cumulative sums of random variables. *Ann. Math. Statist.*, 15(3): 283–296, 09 1944. doi: 10.1214/aoms/1177731235. URL http://dx.doi.org/10.1214/aoms/1177731235.

Yue Wang, Xintao Wu, and Donghui Hu. Using randomized response for differential privacy preserving data collection. In *Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference, EDBT/ICDT Workshops 2016, Bordeaux, France, March 15, 2016.*, 2016. URL http://ceur-ws.org/Vol-1558/paper35.pdf.

Stanley L. Warner. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *Journal of the American Statistical Association*, 60(309):63+, March 1965. URL http://dx.doi.org/10.2307/2283137.

Huasen Wu and Xin Liu. Double thompson sampling for dueling bandits. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 649–657. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6157-double-thompson-sampling-for-dueling-bandits.pdf.

Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. Pebl: Positive example based learning for web page classification using svm. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 239–248, New York, NY, USA, 2002. ACM. doi: 10.1145/775047.775083. URL http://doi.acm.org/10.1145/775047.775083.

Y. Yue and T. Joachims. Beat the mean bandit. In *ICML 2011*, pages 241–248. Omnipress, 2011.

Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, 78(5):1538–1556, 2012.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML 2009*, pages 1201–1208. Omnipress, 2009. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553527.

Bangzuo Zhang and Wanli Zuo. Learning from Positive and Unlabeled Examples: A Survey. In *2008 International Symposiums on Information Processing*, volume 0, pages 650–654, May 2008. URL http://dx.doi.org/10.1109/isip.2008.79.

Dell Zhang. A simple probabilistic approach to learning from positive and unlabeled examples. In *In Proc. of the 5th Annual UK Workshop on Computational Intelligence*, 2005.

Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artif. Intell. Rev.*, 22(3):177–210, November 2003. ISSN 0269-2821. doi: 10.1007/s10462-004-0751-8. URL http://dx.doi.org/10.1007/s10462-004-0751-8.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 928–936, 2003a. URL http://www.aaai.org/Library/ICML/2003/icml03-120.php.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 928–935. AAAI Press,

2003b. ISBN 1-57735-189-4. URL http://dl.acm.org/citation.cfm?id=3041838.3041955.

Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten de Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *ICML 2014*, volume 32 of *JMLR Proceedings*, pages 10–18, 2014a.

Masrour Zoghi, Shimon A. Whiteson, Maarten de Rijke, and Remi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 73–82, New York, NY, USA, 2014b. ACM. ISBN 978-1-4503-2351-2. doi: 10.1145/2556195.2556256. URL http://doi.acm.org/10.1145/2556195.2556256.

Masrour Zoghi, Shimon A Whiteson, Maarten de Rijke, and Remi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *WSDM 2014*, pages 73–82. ACM, 2014c.

Masrour Zoghi, Zohar S Karnin, Shimon Whiteson, and Maarten de Rijke. Copeland dueling bandits. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 307–315. 2015a. URL http://papers.nips.cc/paper/6023-copeland-dueling-bandits.pdf.

Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. Mergerucb: A method for large-scale online ranker evaluation. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 17–26, New York, NY, USA, 2015b. ACM. ISBN 978-1-4503-3317-7. doi: 10.1145/2684822.2685290. URL http://doi.acm.org/10.1145/2684822.2685290.

Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. MergeRUCB: A method for large-scale online ranker evaluation. In *WSDM 2015*. ACM, 2015c.