University of Alberta

The Role of Information in Online Learning

by

Gábor Bartók

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Fall 2012
Edmonton, Alberta

Abstract

In a partial-monitoring game a player has to make decisions in a sequential manner. In each round, the player suffers some loss that depends on his decision and an outcome chosen by an opponent, after which he receives "some" information about the outcome. The goal of the player is to keep the sum of his losses as low as possible. This problem is an instance of online learning: By choosing his actions wisely the player can figure out important bits about the opponent's strategy that, in turn, can be used to select actions that will have small losses. Surprisingly, up to now, very little is known about this fundamental online learning problem.

In this thesis, we investigate this problem. In particular, we investigate to what extent the information received influences the best achievable cumulative loss suffered by an optimal player. We present algorithms that have theoretical guarantees for achieving low cumulative loss, and prove their optimality by providing matching, algorithm independent lower bounds. Our new algorithms represent new ways of handling the exploration-exploitation trade-off, while some of the lower bound proofs introduce novel proof techniques.

Acknowledgements

The past five years at the University of Alberta have been a great experience for me. First of all, I thank my supervisor, Csaba Szepesvári, for giving me the opportunity to come here, for being my mentor while I was his student, for teaching me everything related to doing research.

I thank my main collaborator, Dávid Pál, for working together on this project, for all the meetings and discussion. I thank to all my other collaborators for their help and support: András Antos, Sandra Zilles, and Navid Zolghadr. This thesis could not have been written without your help! I would also like to thank my examining committee for their comments on how to improve the thesis: Shai Ben-David, Michael Bowling, Edit Gombay, Russel Greiner, and Dale Schuurmans. Furthermore, I thank Anna Koop for proofreading the thesis and helping with the writing style.

Finally but most importantly, I would like to thank my family for their support. My wife: Ágnes Pákozdi, and my two daughters: Noémi and Tímea. You guys fill my life with love, happiness, and excitement. The world would be a boring place without you!

Table of Contents

[1] Versions of the work in this chapter appeared in Bartók, Pál, and Szepesvári [2010] and Antos, Bartók, Pál, and Szepesvári [2012].

[2]Based on joint work with András Antos and Csaba Szepesvári.

[3]A version of the work in this chapter appeared in Bartók, Pál, and Szepesvári [2011]

[4]Part of this chapter is based on the work by Bartók, Zolghadr, and Szepesvári [2012] to be published at ICML2012.

[5]Joint work with Navid Zolghadr, Russ Greiner, and Csaba Szepesvári.

List of Figures

List of contributions

1. Classification theorems
   (a) Classification of finite partial-monitoring games with two outcomes for non-stochastic environments. (Theorem 2)
   (b) Classification of finite partial-monitoring games with two actions for non-stochastic environments. (Theorem 8)
   (c) Classification of finite partial-monitoring games for stochastic environments. (Theorem 9)
2. Algorithms
   (a) AppleTree: for finite partial-monitoring games with two outcomes, assuming that the *separation condition* holds. (Section 4.4.1)
   (b) Balaton: for finite partial-monitoring games, assuming that the *local observability condition* holds. (Section 6.2.1)
   (c) CBP-Vanilla: An improvement of Balaton. (Section 7.1)
   (d) CBP: A further improvement of Balaton for finite partial-monitoring games. (Section 7.2)
3. Upper bounds
   (a) 0 regret bound for trivial games. (Lemma 1)
   (b) $\widetilde{O}(\sqrt{T})$ high probability regret bound for AppleTree. (Theorem 3)
   (c) $O(\sqrt{T})$ high probability regret bound for non-hopeless two-action games. (Theorem 7)
   (d) $\widetilde{O}(\sqrt{T})$ expected regret bound for Balaton. (Theorem 10)
   (e) $O(\log T)$ individual expected regret bound for CBP-Vanilla. (Theorem 12)
   (f) $\widetilde{O}(\sqrt{T})$ minimax expected regret bound for CBP-Vanilla. (Corollary 1)
   (g) Individual expected regret bound for CBP. (Theorem 13)
   (h) $\widetilde{O}(T^{2/3})$ minimax expected regret bound for CBP. (Corollary 2)

(i) $\widetilde{O}(\sqrt{T})$ minimax expected regret bound for CBP run against benign opponents. (Theorem 14)

(j) $\widetilde{O}(\sqrt{T})$ minimax expected regret bound for online probing with free labels. (Theorem 15)

(k) $\widetilde{O}(T^{2/3})$ minimax expected regret bound for online probing with costly labels. (Theorem 16)

4. Lower bounds

(a) $\Omega(\sqrt{T})$ expected regret bound on non-trivial finite games. (Theorem 4)

(b) $\Omega(T^{2/3})$ expected regret bound on two-outcome games for which the separation condition does not hold. (Theorem 5)

(c) A simple proof of $\Omega(T)$ expected regret bound for finite hopeless games. (Section 4.2)

(d) $\Omega(T^{2/3})$ expected regret bound on finite games with no local observability, against non-stochastic opponents. (Theorem 11)

(e) $\Omega(T^{2/3})$ expected regret bound on finite games with no local observability, against stochastic opponents. (Theorem 11)

(f) $\Omega(T^{2/3})$ expected regret bound on online probing with costly label. (Theorem 17)

Chapter 1

Introduction

In this chapter we first introduce the full- and bandit-information versions of prediction with expert advice. Next, we present the framework of partial monitoring, which generalizes both of the above. Then, we define the notion of learnability and regret of a prediction problem. The material presented in this chapter is a standard part of the literature, whose most relevant part will be reviewed in Chapter 2.

1.1 Prediction with expert advice: full and bandit information

Online learning is a problem formulation in machine learning where a learner has to make decisions on a turn-by-turn basis. On every turn, after the learner makes his decision, he suffers some loss.[1] The loss suffered depends on the learner's decision and some unknown process running in the background. Before making his decision, the learner might receive some additional information about the current turn. Unlike in other learning models such as supervised learning, the learner is evaluated based on losses suffered *during* the learning process. Consider the following example:

Example 1. Temperature forecasting. Every day you must predict the temperature for the next morning. Your loss is the absolute difference between your prediction and the actual temperature.

Many mass phenomena can be modelled as a stochastic process. In the case of temperature forecasting, one possibility is to view the sequence of temperatures as an i.i.d.[2] sequence, in which case one can prove that a good strategy would be to predict the empirical median of past observations. One may however wonder if predicting the median is always a good strategy? In particular, what happens when the i.i.d. assumption is violated? Will the

[1] For simplicity we chose to adopt the masculine form and also because most of our learners are imperfect as are our masculine brothers.

[2] Independent, identically distributed.

prediction strategy break down in an uncontrolled manner? In particular, how should one evaluate a given prediction strategy? An attractive approach, which allows the generalization of many results available for the stochastic case is to evaluate the forecaster by comparing his total loss to the loss of each of a fixed set of competitors.

In many cases, the predictions of the competitors can actually be used to come up with one's own forecast. This leads to the problem of prediction with expert advice.

Example 2. Temperature forecasting with expert advice. Assume that you are the CEO of a weather-forecast company, which is paid based on how accurately it predicts the temperature for the next morning. You hire $N \in \mathbb{N}$ professional forecasters (experts) to help you. Every day at noon, the experts send their temperature predictions for the next morning to you. With the help of data collected about how well the individual experts could predict the temperature in the past, you should decide what temperature to predict for tomorrow morning. The next morning you get to know the actual temperature. Your loss is the absolute difference between your prediction and the real temperature. If you knew which of your experts is going to be most accurate, you could just take his prediction: it is assumed that the loss of this expert is small enough. Thus, the goal is to compete with this best expert in hindsight.

It has been proven that, in this setting, one can predict "almost as well" as the best expert in the sense that there exist an algorithm such that, using this algorithm, the average excess loss of the forecaster compared to the best expert will vanish in the long run.[3]

In the above example, the outcome (*i.e.,* the morning temperature) is released every day. This means that the learner has access to *full information* in the sense that he can evaluate the losses of all the experts (and in fact, every possible prediction) every morning. However, in some problems the information received by the learner is restricted. Imagine for example that in the temperature forecasting problem the temperature is not released but only the difference between the actual temperature and the prediction becomes available every morning. Although this problem looks contrived, in many other cases there is no other choice than to assume this form of restricted feedback:

Example 3. Multi-armed bandits. You go to a casino with $N \in \mathbb{N}$ slot machines. Every slot machine works differently. In every time step, you pull an arm of one of the slot machines and receive a reward (or suffer a loss). The problem is to collect as much reward as possible. You are evaluated in comparison to the best possible arm in hindsight.

In this example, if you pull an arm in a time step, you do not know what would have happened had you pulled a different arm. Therefore, there are two competing strategies serving different needs:

[3]The precise statement, together with references, will be given in the next chapter.

1. Pull the arm that looks the best.

2. Pull an arm that has not been pulled too much.

The first pure strategy attempts to *exploit* the knowledge gathered so far, whereas the second one attempts to *explore* to gain more information. It is easy to see that we need to use a compound strategy that mixes the two strategies: if we only exploit, we might miss some good arms that looked bad at the beginning, but if we always explore we might rarely gain high rewards. The problem of how to mix these two pure strategies is called the *exploration-exploitation dilemma* and is a recurring feature of many online learning problems.

Examples 2 and 3 illustrate the two most widely used feedback models in online learning. However, there exist several online learning problems that cannot be modeled as either of these. A general framework that allows the modeling of such problems is the framework of *partial monitoring*.

1.2 Partial monitoring

In *partial monitoring* learning problems, a *player* and an *opponent* play a repeated game. The game, $\mathcal{G} = (\mathcal{N}, \mathcal{M}, \Sigma, \mathcal{L}, \mathcal{H})$, is specified by an information set Σ, an action set $\mathcal{N}$, an outcome set $\mathcal{M}$, a loss function $\mathcal{L} : \mathcal{N} \times \mathcal{M} \mapsto \mathbb{R}$, and a feedback function $\mathcal{H} : \mathcal{N} \times \mathcal{M} \mapsto \Sigma$. In every round, the opponent and the player simultaneously choose an *outcome* J_t from $\mathcal{M}$ and an *action* I_t from $\mathcal{N}$, respectively. The player then suffers the loss $\ell_t = \mathcal{L}(I_t, J_t)$ and receives the feedback $h_t = \mathcal{H}(I_t, J_t)$. Only the feedback is revealed to the player, the outcome and the loss remain hidden. The player's goal is to minimize his cumulative loss and his performance is measured against the best action in hindsight: The difference between the best possible loss that could have been achieved and the actual loss of the player will be called the *regret*. We assume that the range of the losses is bounded[4], typically in $[-1, 1]$ or $[0, 1]$. Note that this model assumes no noise: the loss and feedback are deterministic given the action and the outcome. It is also important to note that the game (and in particular, the functions $\mathcal{L}$ and $\mathcal{H}$) are revealed to the player before the game begins.

Because we wish to allow stochastic choices of actions and outcomes in the sequel, we shall assume that $\mathcal{N}$ and $\mathcal{M}$ are measurable spaces. In most of our examples, in fact these spaces will be trivially measurable since they are finite. In the few other remaining examples, these spaces will be subsets of appropriate Euclidean spaces. For these cases we will consider Borel measurability.

Based on the opponent's strategy for selecting outcomes, we distinguish three cases:

[4]Although there exist some results for unbounded losses (see *e.g.,* Allenberg et al. [2006]), most of the related literature makes the assumption that losses are bounded.

- *Stationary memoryless, stochastic* opponent: The outcomes are chosen in an i.i.d. manner from a distribution defined over $\mathcal{M}$.

- *Oblivious adversarial* opponent: The opponent chooses an outcome arbitrarily at every time step. The player's actions are never revealed to the opponent and thus we can equivalently think of the opponent as an arbitrary sequence of outcomes that is fixed before the game begins.[5]

- *Non-oblivious* or *adaptive adversarial* opponent: At every time step, the opponent chooses an outcome arbitrarily. In this case the opponent has access to the player's past actions.

In this thesis we only deal with opponents of the first two types. From now on, the term "adversarial opponent" refers to the oblivious adversarial case.

While the opponent is not allowed to make use of the player's previous actions when choosing an outcome, the player can, and should, use his past observations. A *strategy* or *algorithm* $\mathcal{A}$ is a function[6] that outputs an action at every time step t based on the history $H_{t-1} = (I_1, h_1, \ldots, I_{t-1}, h_{t-1})$ and a random variable ξ_t, where $(\xi_1, \xi_2, \ldots)$ is an i.i.d. sequence of uniformly distributed random variables. The ability to randomize the player's decisions is essential when playing against an adversarial opponent: deterministic algorithms might be second-guessed, causing high regret[7]. We denote by $\mathcal{A}(H_{t-1}, \xi_t)$ the "decision" of the algorithm. Since one typically wants to design algorithms for a class of games instead of a single game, $\mathcal{A}(H_{t-1}, \xi_t)$ is an abuse of notation. The algorithm makes its decision based not only on the history and the randomness but it receives, as mentioned earlier, the game $\mathcal{G}$ itself.

1.3 Examples

In this section we describe some "real-world" examples of learning problems that can be modeled in the partial monitoring framework. For this, we need to specify the elements of the tuple $\mathcal{G} = (\mathcal{N}, \mathcal{M}, \Sigma, \mathcal{L}, \mathcal{H})$.

Example 4. Horse race. Suppose there is a horse race every day with the same $N \in \mathbb{N}$ horses. The goal is to predict the winner before each race. After each race, the results, an ordering of the horses, is announced. If the horse we predicted to win comes k^{th}, the loss is $(k-1)/N$.

[5] Technically, the choices of the opponent could still be stochastic.

[6] The word "algorithm" is misused in the sense that computational aspects are not discussed. Further, in what follows we will essentially identify learners with the algorithm that they use, so the words algorithm and learner will be used interchangeably.

[7] Deterministic algorithms can achieve low regret under some special conditions even against an adversary. For example, if the action space is some convex set in a Euclidean space and the loss function is strongly convex for any outcome, then, in the case of $\mathcal{L} = \mathcal{H}$, i.e., full-information problems, a very simple strategy called "Follow the leader" can achieve logarithmic regret in terms of the time horizon.

This problem is put into our framework as follows: The player has to choose a horse, thus $\mathcal{N} = \{1, \ldots, N\}$. An outcome is a permutation of the horses, hence $\mathcal{M} = perm(\mathcal{N})$. In particular, for $\pi \in \mathcal{M}$, $\pi : \mathcal{N} \to \mathcal{N}$ is a bijection and $\pi(i)$ will denote the place of horse i in the outcome. The feedback is the same as the outcome, thus $\Sigma = \mathcal{M}$ and $\mathcal{H}(i, \pi) = \pi$, where $i \in \mathcal{N}$, $\pi \in \mathcal{M}$. Finally, $\mathcal{L}(i, \pi) = (\pi(i) - 1)/N$.

This example is an instance of a full information game because the losses of all actions can be recovered based from the feedback no matter the outcome and the action:

Definition 1. *A partial-monitoring game $\mathcal{G} = (\mathcal{N}, \mathcal{M}, \Sigma, \mathcal{L}, \mathcal{H})$ is a game with* full information *if there exists a function $f : \mathcal{N} \times \Sigma \mapsto \mathbb{R}^{\mathcal{N}}$ such that for any action $i \in \mathcal{N}$ and outcome $j \in \mathcal{M}$,*

$$f\left(i, \mathcal{H}(i, j)\right) = \mathcal{L}\left(\cdot, j\right) .$$

Example 5. Commuting. We go to our workplace every morning. We can choose to do so by car, public transport, or bike or we can walk, or ask our colleague to pick us up. The loss is the time spent on commuting.

The action set is the five ways of going to work ($\mathcal{N} = \{1, 2, \ldots, 5\}$) the outcome is the losses for all possibilities, thus, assuming that each individual loss is in the $[0, 1]$ interval, $\mathcal{M} \subset [0, 1]^5$. The loss function is $\mathcal{L}(i, x) = x_i$, where $x = (x_1, \ldots, x_5) \in \mathcal{M}$. The feedback is a loss, thus, $\Sigma = [0, 1]$. Our feedback, however, is restricted to the loss for the action we chose: $\mathcal{H}(i, x) = x_i$. This game is an instance of what we call a *bandit game*, or a game with *bandit information*. These games allow the player to recover the loss of the action chosen from the feedback received:

Definition 2. *A partial-monitoring game $\mathcal{G} = (\mathcal{N}, \mathcal{M}, \Sigma, \mathcal{L}, \mathcal{H})$ is a game with* bandit information *if there exists a function $f : \mathcal{N} \times \Sigma \mapsto \mathbb{R}$ such that for any action $i \in \mathcal{N}$ and outcome $j \in \mathcal{M}$,*

$$f\left(i, \mathcal{H}(i, j)\right) = \mathcal{L}\left(i, j\right) .$$

Note that this definition does not state that the only information given by the feedback is the loss of the action we chose. For example, according to this definition, all full information games are bandit information games as well.

Example 6. Dynamic pricing. Consider a monopolist who has an unlimited supply of a nonperishable single product, with no marginal cost of production. The monopolist (or seller) can set the prices for the product and it is assumed that he will receive feedback in real time for each individual customers. Each customer (buyer) secretly decides about a maximum price he is willing to pay for the product. If the buyers's maximum price is lower than the seller's price, no transaction happens and the seller suffers some constant storage loss. Otherwise, the transaction occurs and the seller's loss is the difference between the buyer's maximum price and his own price.

In this example, the learner is the seller and the opponent is the buyer, $\mathcal{N}, \mathcal{M} \subset \mathbb{R}$, $\Sigma = \{\text{“sold”}, \text{“not sold”}\}$, and the loss and feedback functions can be described as

$$\mathcal{L}(i,j) = \begin{cases} c, & \text{if } i > j; \\ j - i, & \text{if } i \leq j, \end{cases}$$
$$\mathcal{H}(i,j) = \begin{cases} \text{“not sold”}, & \text{if } i > j; \\ \text{“sold”}, & \text{if } i \leq j, \end{cases}$$

where c is a fixed cost of lost sales. Interestingly, this problem cannot be modeled as either a full information game or a bandit game. In particular, it is not possible to recover the loss of the action chosen from the feedback received. Thus, at first sight, it may be surprising that it is still possible for the player to act in a reasonable manner.

1.4 Learnability and regret

As briefly mentioned previously, the performance of the learner is evaluated based on his cumulative loss as compared to that of the best fixed action in hindsight over some period of time of length T. Formally, this difference or *regret* is defined by

$$R_T = R_T^{\mathcal{A}}(J_1, \ldots, J_T) = \sum_{t=1}^{T} \mathcal{L}(I_t, J_t) - \min_{i \in \mathcal{N}} \sum_{t=1}^{T} \mathcal{L}(i, J_t)\,.$$

Note that in this definition, the regret depends on the choices of both the learner and the opponent, and might even be random if, for example, the learner (or his opponent) randomizes his actions.

A question of major importance is how the regret depends on the length of the time horizon T. For example, if the regret grows linearly with T then this means that the learner performs significantly worse than the best constant action, and we say that the learner fails to learn. On the other hand, if the regret is sublinear in T, $R_T = o(T)$ or $\limsup_{T\to\infty} R_T/T = 0$ (in an appropriate probabilistic sense) then this means that in the long run the learner's average loss gets infinitesimally close or better than that of the best action in hindsight. In this case, we may say that the learner “learned to play” the game.

In case of a sublinear regret, it becomes important to have a closer look at the growth rate of the regret, as it can tell us “how efficiently” does the learner learn.

For the different opponent models, we measure the performance of the learner in slightly different ways. These definitions, given in the next two sections, differ in terms of how we deal with the stochastic nature of the regret.

1.4.1 Regret against adversarial opponents

Consider online learning against an adversarial opponent. In this case, we will be interested in the *worst-case* regret of an algorithm $\mathcal{A}$ defined as

$$R_T^{\mathcal{A}}(\mathcal{G}) = \sup_{J_1,\dots,J_T \in \mathcal{M}} R_T^{\mathcal{A}}(J_1, \dots, J_T)\,.$$

The worst-case regret, as follows from its name, describes how well the algorithm does when the opponent uses the "hardest" possible outcome sequence against the algorithm.

The *minimax expected regret* (or *minimax regret* for short) builds on top of the worst-case regret and it indicates the "hardness" of a game $\mathcal{G}$ itself:

$$R_T(\mathcal{G}) = \inf_{\mathcal{A}} \mathbb{E}[R_T^{\mathcal{A}}(\mathcal{G})]\,.$$

Intuitively, the minimax regret is the worst-case expected regret of the best possible algorithm.

1.4.2 Regret against stochastic opponents

The notion of regret is slightly relaxed for stochastic opponents in that instead of comparing with the cumulative loss of the best action in hindsight, we compare against the expected cumulative loss of the action with the smallest expected loss. Formally, let μ be a distribution over $\mathcal{M}$ and let the outcomes $(J_t)_{1\le t\le T}$ form an i.i.d. sequence with the common marginal μ (*i.e.,* $J_t \sim \mu$). Then, we define the regret of algorithm $\mathcal{A}$ against μ as

$$R_T^{\mathcal{A}}(\mu) = \sum_{t=1}^{T} \mathcal{L}(I_t, J_t) - T \inf_{i\in\mathcal{N}} \mathbb{E}[\mathcal{L}(i, J_1)]\,.$$

We also call this type of regret the *individual* or *problem dependent* regret. In the same manner as in the previous case, we define the minimax expected regret of a game:

$$R_T(\mathcal{G}) = \inf_{\mathcal{A}} \sup_{\mu} \mathbb{E}[R_T^{\mathcal{A}}(\mu)]\,.$$

Note that we have heavily overloaded the notation R_T. However, from the context it should always be clear if we mean regret against a stochastic or an adversarial opponent, whereas the arguments $R_T(\cdot)$ make it clear if we mean minimax, worst-case, or individual regret.

1.4.3 Bounds on the regret

In the previous sections we defined our two regret concepts. In this section, we discuss the form of bounds on the regret that we will prove in the subsequent chapters.

Upper bounds on the regret. If we design an algorithm $\mathcal{A}$, we want to know how it performs on a game.[8] To this end, we prove upper bounds on the regret. Let us first discuss the case of worst-case regret against adversarial opponents. Since the regret can be random, we usually look at the following kinds of bounds:

1. *High probability* worst-case bounds. These bounds usually state that no matter what the game is (within a selected class), for any given $0 \leq \delta < 1$, the regret satisfies with probability at least $1 - \delta$,

$$R_T^{\mathcal{A}}(\mathcal{G}) \leq f(\mathcal{G}, T, \delta)$$

   with some function f.

2. *Expected* worst-case bounds. As the name suggests, in such a bound the expected worst-case regret is bounded:

$$\mathbb{E}[R_T^{\mathcal{A}}(\mathcal{G})] \leq f(\mathcal{G}, T)\,.$$

When the opponent is stochastic, we have the analogue bounds, but we allow the function f to depend on μ, the outcome distribution used by the opponent.[9] Bounds that do not depend on μ are worst-case against stochastic opponents. Such bounds are also called *uniform*.

Lower bounds on the minimax regret. These results give lower bounds on the minimax regret of a game, against stochastic or adversarial opponents. These bounds become important when one wants to decide if an algorithm is (near-)optimal for a game; if we have an upper bound on our algorithm for the worst-case regret, and we also have a lower bound on the minimax regret then, by comparing the two bounds, we can tell if the algorithm is further improvable (in a worst-case sense).

In this work, we focus in particular on how the minimax regret scales with the time horizon. In particular, we will say that the minimax regret is of order $O(T^\alpha)$ up to logarithmic factors if there exists an algorithm with worst-case regret scaling as $O(T^\alpha \mathrm{polylog} T)$, while at the same time a lower bound for the minimax regret scaling as $\Omega(T^\alpha)$ is also known.

In the subsequent chapters we will investigate partial-monitoring games of different types. The main problem will be to determine the minimax regret of any given game.

[8] Here, the algorithm is meant in the general sense of a method that can be applied to many games. In particular, the algorithm may "read" the definition of the game and modify its behavior based on the game's definition.

[9] It is possible to derive refined bounds in the adversarial case, as well, that depend on the outcomes chosen by the opponent [*e.g.,* Hazan and Kale, 2008, 2011]. However, in this thesis we will not consider such bounds.

1.5 Relation to supervised learning

In the field of supervised learning, the learner is given a set of *labeled examples* $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ sampled i.i.d. from a distribution, where x_i are *feature vectors*, and y_i are *labels*. The task of the learner is to present a decision maker that maps feature vectors to labels. The learner is evaluated based on how accurately it is able to predict the label, given a feature vector x sampled (usually) from the same distribution as the training examples.

The online version of the game can be formulated as a repeated game where at time step t, x_t is given to the learner, he guesses a label $\hat{y}_t$ and then receives the true label y_t.

The main difference between the two versions of the problem is that while in the supervised learning setting the learner is evaluated based on his accuracy *after* seeing all training examples, in the online learning setting the learner is evaluated based on his accuracy *during the learning process*. This makes the online learning setting inherently harder.

In fact, it is shown that for any online learning game, any learner with small regret can be turned into a supervised learning agent that achieves good accuracy. This transformation is called "Online to batch conversion" (see *e.g.,* Littlestone [1989], Cesa-Bianchi et al. [2004], Dekel and Singer [2006]). The simplest of these methods works as follows: At every time step t, the online learner has a decision maker that predicts y_t. We let the supervised learner choose a decision maker uniformly randomly from this set.

The precise statement, taken from Cesa-Bianchi et al. [2004] is as follows.

Theorem 1. *Let $\mathcal{D}$ be the decision space of the predictor and $\mathcal{Y}$ be the set of labels. Let $\mathcal{D}$ be convex and the loss function $\ell : \mathcal{D} \times \mathcal{L} \mapsto [0, L]$ be convex in its first argument. Let an arbitrary online algorithm output hypotheses $H_0, \ldots, H_n$ when run on examples $(X_1, Y_1), \ldots, (X_n, Y_n)$. Let $\mathrm{er}(h) = \mathbb{E}[\ell(h(X), Y)]$ where (X, Y) is drawn from the same distribution as the examples. Then, for any $0 < \delta \leq 1$, the hypothesis $\bar{H} = 1/n \sum_{t=0}^{n-1} H_t$ satisfies*

$$\mathbb{P}\left(\mathrm{er}(\hat{H}) \geq \frac{M}{n} + L\sqrt{\frac{2}{n}\log\frac{1}{\delta}}\right) \leq \delta\,,$$

where M is the cumulative loss of the online algorithm.

Chapter 2

Background

In this chapter we provide a more detailed background of online learning, necessarily restricting ourselves to literature highly relevant to our research topic given the enormity of the field.

2.1 Full-information games

Sequential decision making problems have been studied since at least the 1950s [Blackwell, 1954, Hannan, 1957]. Most of the early works addressed the full-information case with finitely many actions and outcomes. Perhaps the best known algorithm for this problem is due to Littlestone and Warmuth [1994] and Vovk [1990]. Their method, called the *Weighted Majority* algorithm (or aggregating algorithm by Vovk), maintains weights for the actions, uses a multiplicative update based on the observed losses and draws an action at every time step from the distribution defined by the weights (see Algorithm 4.4.1).

This Weighted Majority algorithm, with appropriately set parameter, is known to achieve an expected regret of $\sqrt{(T/2)\log N}$, which is optimal in the sense that there is a matching lower bound. As we will see in examples below, this algorithm is the core of the algorithms designed for many other online learning problems that assume adversarial opponents.

2.2 Multi-armed bandits

The multi-armed bandit problem with stochastic opponent was first introduced by Robbins [1952]. For the case where the opponent's distributions come from a smoothly parameterized family of distributions, a solution enjoying unimprovable asymptotic individual regret for a large family of problems was presented by Lai and Robbins [1985]. Their solution for some sufficiently regular families of parametric distributions can be summarized as follows:[1]

[1]In fact, Lai and Robbins [1985] gave an algorithm schema that is built around general, but unspecified estimators of the mean and the so-called upper confidence indices. At the

Algorithm 1 The **Weighted Majority** algorithm (after Littlestone and Warmuth [1994] and Vovk [1990])

Input parameter: $\eta > 0$
Initialization: $w \leftarrow (1, \ldots, 1)$
for $t = 1, 2, \ldots$ **do**
  Play action I_t drawn from the multinomial distribution with parameters $p_i = w_i / \sum_{j=1}^{N} w_j$, $i = 1, \ldots, N$
  Observe J_t
  Compute the losses $\ell_i \leftarrow \mathcal{L}(i, J_t)$, $i = 1, \ldots, N$
  Update the weights $w_i \leftarrow w_i \exp(-\eta \ell_i)$, $i = 1, \ldots, N$
end for

1. Choose each action once.

2. From time step $K + 1$, calculate maximum likelihood based estimates of the parameters for all the distributions underlying each action and also maintain an *upper confidence bound*. This upper confidence bound is calculated based on a Kullback-Leibler ball around the maximum likelihood estimates.

3. At time step $kK + i$ ($k \in \mathbb{Z}^+$), choose action i if its upper confidence bound is larger than $\max_{1 \le i \le K} \hat{\mu}_i$, where $\hat{\mu}_i$ is an estimate of the mean of action i; otherwise choose $\arg\max_{1 \le i \le K} \hat{\mu}_i$.

This algorithm, called *Upper Confidence Indices*, UCI introduces the principle of "optimism in the face of uncertainty". If we do not know "enough" about an action (*i.e.,* we have not chosen it sufficiently many times), the action is assumed to be a good action and we choose it. This method ensures that every action will be explored sufficiently frequently.

The general UCI algorithm needs to find the upper confidence indices numerically. In general, this means that it needs to store the whole history of observations up to the current time step, which might be problematic in some application. This issue was overcome by Agrawal [1995], who, instead of using all the samples, calculated the upper confidence bounds based only on the sample means and the counts of the number of times the actions were chosen.

These results were generalized to the non-parametric case by Auer et al. [2002], who introduced the algorithm UCB1. In UCB1, the upper confidence bounds are computed in a simple way based only on the sample means and the arm counts, and the action chosen in every time step is the one that has the highest upper confidence bound. They also prove a finite time regret bound of $O(N \log T)$, where the $O(\cdot)$ notation hides problem dependent parameters, namely the differences between the mean of the optimal arm and that of the suboptimal arms.

price of losing generality, we decided to simplify their general solution schema, while keeping its essence, to make it easier to follow.

Algorithm 2 UCB1 due to Auer et al. [2002]. Formulated using rewards.

Initialization: Choose each action once
for $t = N+1, N+2, \ldots$ **do**
  $I_t \leftarrow \arg\max_{i=1,\ldots,N} \left(\bar{x}_i + \sqrt{\frac{2 \log t}{n_i}} \right)$ {$\bar{x}_i$: average reward for action i, n_i: number of times action i was chosen}
  Receive reward
  Update $\bar{x}_{I_t}$ and n_{I_t}
end for

Algorithm 3 The **Exp3** algorithm [Auer et al., 2003]

Input parameters: $\gamma > 0, \eta > 0$
Initialization: $w \leftarrow (1, \ldots, 1)$
for $t = 1, 2, \ldots$ **do**
  Play action I_t drawn from the multinomial distribution with parameters $p_i = (1-\gamma)\frac{w_i}{\sum_{j=1}^{N} w_j} + \gamma \frac{1}{N}$, $i = 1, \ldots, N$
  Observe loss $\ell_{I_t} \leftarrow \mathcal{L}(I_t, J_t)$
  Compute $\hat{\ell}_{I_t} \leftarrow \ell_{I_t} / p_{I_t}$
  Update weight $w_{I_t} \leftarrow w_{I_t} \exp(-\eta \hat{\ell}_{I_t})$
end for

The multi-armed bandit problem against an adversarial opponent was analyzed by Auer et al. [2003], who introduced a variety of algorithms for the non-stochastic multi-armed bandit problem and proved regret bounds with different requirements. The simplest of them, called *Exponential-weight algorithm for Exploration and Exploitation*, or Exp3, is proven to achieve expected regret[2] of $\widetilde{O}(\sqrt{NT})$. This matches a lower bound, proven in the same article, up to a logarithmic factor in the number of arms.

The Exp3 algorithm builds on the Weighted Majority algorithm. There are two main differences between these algorithms. First, when Exp3 draws an action, it does not directly use the weights, but mixes them with the uniform distribution. This is to make sure that the algorithm explores all the actions sufficiently frequently, even the ones that seem suboptimal. Second, since Exp3 does not have access to all the losses, it uses an estimate at every time step. The estimate $\hat{\ell}_i$ is defined to be zero whenever i is not the chosen action and $\hat{\ell}_i = \ell_i / p_i$ when i is the chosen action, where p_i is the probability of choosing action i. It can be shown that the estimates $\hat{\ell}_i$ are unbiased estimates of the real losses at every time step, given the history.

In their recent work, Audibert and Bubeck [2010] removed the logarithmic factor from the upper bound of the minimax regret of bandit games. Their algorithm, INF for Implicitly Normalized Forecaster, is presented in Algorithm 4. The algorithm is similar in spirit to Exp3, but instead of exponential weights, they use a cleverly chosen potential function. One drawback of INF

[2]The notation $\widetilde{O}(\cdot)$ hides polylogarithmic terms.

Algorithm 4 The INF algorithm [Audibert and Bubeck, 2010]

Input parameters: function $\psi : \mathbb{R}_-^* \mapsto \mathbb{R}_+^*$
Initialization: $p \leftarrow (1/N, \ldots, 1/N)$, $\hat{G}_i \leftarrow 0$ $(i = 1, \ldots, N)$
for $t = 1, 2, \ldots$ **do**
  Play action I_t drawn from p.
  Observe gain g_{I_t}
  Compute $\hat{g}_i \leftarrow \frac{\mathbb{I}_{\{I_t=i\}} g_{I_t}}{p_i}$, $\hat{G}_i \leftarrow \hat{G}_i + \hat{g}_i$ for $i = 1, \ldots, N$
  Let C normalization factor satisfy $\sum_{i=1}^N \psi(\hat{G}_i - C) = 1$
  Compute $p_i \leftarrow \psi(\hat{G}_i - C)$ for $i = 1, \ldots, N$
end for

is that it uses an implicit normalization at every time step (hence the name), and thus one has to use some approximation to run the algorithm.

2.2.1 Bandits with experts

In the original bandit model, the regret is defined as the excess loss compared to the best constant action. In the temperature prediction problem (Example 2), the goal is to compete with the best forecaster, which suggests that the set of actions should be identified with the set of experts. However, the problem has more structure to it: the experts all predict temperatures and even the loss is defined in terms of temperatures. We can say that the set of temperatures in this problem form a set of *primitive actions*. This leads to the consideration of a two-level model, where the goal is to compete with the best expert predictor from an expert set, and all experts act by choosing a prediction from the primitive action set $\mathcal{N}$. As before, losses and feedbacks can be assigned to each (primitive) action-outcome pair.

In the full information case, the additional structure is not particularly helpful. However, this is not the case for bandit information: Assuming that the experts announce the probability distributions that they would use to choose the primitive actions, Auer et al. [2003] proposed an algorithm called Exp4 (building on Exp3), which was shown to achieve $O(\sqrt{TN \log K})$ expected regret, where K is the size of the expert set. What is notable here is that the bound depends only on the logarithm of the number of experts. In contrast, if Exp3 was directly applied to the problem, the bound would be $O(\sqrt{TK \log K})$. Thus, when $N \ll K$, the bound of Exp4 is better. More recently, an algorithm which further improves this bound to $O(\sqrt{TS \log K})$ with $S \leq \min(K, N)$ was suggested by McMahan and Streeter [2009]. Here, the parameter S measures the extent to which the experts agree in their recommendations.

2.3 Bandits with infinitely many arms

As we see from the previous section, the minimax regret of the bandit game grows with the number of arms. It follows that, unless we introduce some new assumptions, bandit problems with infinitely many arms are not learnable.

If we assume that the set of arms is a subspace of $\mathbb{R}^d$ for some dimension d and the reward of an arm $v \in \mathbb{R}^d$ is a random variable with mean $\theta^\top v$ for some hidden $\theta \in \mathbb{R}^d$, we arrive at the *stochastic linear bandit* problem. This case was considered by Auer [2003] and later by Dani et al. [2008]. Their algorithms use the upper confidence bound idea. Instead of calculating a confidence interval around the mean reward[3], a confidence ellipsoid around the parameter θ is constructed. Dani et al. [2008] showed that if the set of arms is a polytope then the expected regret can be bounded from above by $O(d^2 \log T)$. However, if the set of arms has a "smooth" surface (*e.g.,* the set is a ball) then no algorithm can achieve a regret smaller than $\Omega(d\sqrt{T})$. Dani et al. also present an algorithm that achieves $O(d\sqrt{T}\log^{3/2} T)$ expected regret. The same bound is achieved by Rusmevichientong and Tsitsiklis [2010] who propose an algorithm that, as opposed to the strategy of Dani et al., does not require the knowledge of the time horizon T. More recently, Abbasi-Yadkori et al. [2011] proposed a new variant that was shown to improve the time-dependence of the bound by removing a multiplicative factor $\log^{1/2} T$ from the bound. At the same time, they have shown that their new algorithm improves the practical performance of the previous algorithms by a large margin.

In the adversarial version of linear bandits, the opponent's outcome space is $\mathcal{M} \subset \mathbb{R}^d$. The reward of an arm v at time step t is $\theta_t^\top v$, where θ_t is the outcome at time step t. This model is called *bandit linear optimization* and was analyzed by Abernethy et al. [2008], who showed that the minimax regret of such a game is $\widetilde{O}(d\sqrt{T})$, whenever the set of arms is convex and compact.

Another possible assumption is to assume the set of arms comes from a metric space and the average reward function is Lipschitz. This model was considered by Kleinberg et al. [2008] who proved that for uniformly locally α-Lipschitz reward functions with $0 < \alpha \leq 1$ over the interval $[0, 1]$ the minimax expected regret is $\widetilde{O}(T^{(1+\alpha)/(1+2\alpha)})$. When $\alpha = 1$, i.e., the function is Lipschitz, this gives a regret of order $\widetilde{O}(T^{2/3})$. This result was extended by Auer et al. [2007] to the case when the reward function enjoys a higher order smoothness around its maxima. In particular, it was shown that if the reward function has finitely many maxima and the function can be well approximated by a quadratic function in the vicinity of its maxima then the minimax regret is $\Theta(\sqrt{T \log T})$. There also exist extensions of these results to the multi-dimensional case (see Kleinberg et al. 2008, Bubeck et al. 2009, 2011

[3]Rewards can be thought of as the negation of losses. There is no consensus amongst researchers about whether to use rewards or losses in online learning. In the bandit literature, rewards are more common, while in other areas of online learning, researchers use losses. In this document, we will use losses, however, in the literature review we keep the one the original source used.

and the references therein).

If we assume that the loss function is convex, the model of *online convex optimization* arises (e.g., Zinkevich 2003). In this model, the outcome set $\mathcal{M}$ is the set of convex functions over a convex and compact subset $\mathcal{N}$ of $\mathbb{R}^d$ (note that $\mathcal{N}$ still plays the role of the action set). This model is very general in the sense that many online learning problems can be cast as online convex optimization. The related literature on this topic is enormous and is beyond the scope of this document. For an introduction, the reader is advised to refer to the book by Cesa-Bianchi and Lugosi [2006] and references therein.

2.4 Between bandits and full-information

The work of Mannor and Shamir [2011] deals with online learning games where the feedback structure is a hybrid between bandits and full-information. Their assumption is that when the learner chooses an action, he observes the loss of that action along with the losses of some other actions. The feedback structure can be represented as a graph: the vertices correspond to the actions, and there is an edge from an action i to action j if, by choosing action i, the learner also observes the loss of action j. In their paper, Mannor and Shamir [2011] distinguish two cases:

1. The undirected case when an edge from i to j implies the existence of the edge from j to i. That is, if an action helps observing the loss of another, it works the other way as well.

2. The directed case, when the above assumption does not necessarily hold.

In their work, Mannor and Shamir introduce new algorithms for both cases, and provide upper bounds for the expected regret.

Their first algorithm, introduced for the undirected case is called ExpBan. This algorithm is a combination of an experts algorithm and a bandit algorithm. First it splits the graph to cliques (complete subgraphs), and define the cliques as "meta-actions". Then the algorithm plays a bandit algorithm on the cliques while within each clique, it chooses an action using a full-information algorithm. The expected regret of this algorithm heavily depends on the number of cliques needed to partition the graph. Denoting the minimum number of cliques needed by χ, the expected regret of this algorithm is shown to be bounded by

$$\mathbb{E}[R_T] \leq C\sqrt{T\chi \log N}\,.$$

This result is quite satisfying in the sense that it interpolates between the bandit and the full-information upper bound. Indeed, for bandits, $\chi = N$ gives back the result of Exp3, while for full-information games, $\chi = 1$ leads to the bound of the Weighted Majority algorithm.

Algorithm 5 ELP (taken from Mannor and Shamir [2011])

Input: $\eta, \{\gamma(t)\}, \{s_i(t)\}$, neighbor sets of actions $\{N_i(t)\}$
Initialization: $w \leftarrow (1, \ldots, 1)$
for $t = 1, \ldots, T$ **do**
  Play action I_t drawn from the multinomial distribution $p = (p_1, \ldots, p_N)$ with $p_i = (1 - \gamma(t))\frac{w_i}{\sum_{j=1}^{N} w_j} + \gamma(t)s_i(t)$, $1 \le j \le N$
  Observe rewards g_j where $j \in N_{I_t}(t) \cup \{I_t\}$
  Compute the reward estimates $\tilde{g}_j \leftarrow \frac{\mathbb{I}_{\{I_t \in N_j(t)\}}}{\sum_{i \in N_j(t)} p_i} g_j$, $1 \le j \le N$
  Update the weights $w_j \leftarrow w_j e^{\eta \tilde{g}_j}$, $1 \le j \le N$
end for

The other algorithm introduced by Mannor and Shamir is the ELP algorithm (for "Exponentially-weighted algorithm with Linear Programming"), whose pseudocode is given as Algorithm 5. This algorithm works also for problems where the graph is different in every time step. The algorithm builds on Exp3 with the twist that the exploration distribution is calculated via a clever linear programming problem, whose solution is the values $s_i(t)$ (a similar trick is used by McMahan and Streeter 2009). The upper bound derived for ELP is stronger than that of ExpBan for the undirected case: instead of the clique-covering number, the *independence number* (the maximum number of vertices that do not have edges between them) appears in the bound. The paper also shows a matching lower bound for the undirected case. For the directed case, the bound is identical to that of ExpBan on the undirected case.

2.5 Finite partial monitoring

Finite partial monitoring is a special case of partial monitoring. Here the action set $\mathcal{N}$ and the outcome set $\mathcal{M}$ are finite. Hence, the loss function and the observation function can be represented as two matrices, one for the values of the loss function for each pairs of actions and outcomes and one for the values of the feedback function. A finite partial monitoring game is defined by a pair of N-by-M matrices $(\mathbf{L}, \mathbf{H})$, where N and M are the number of actions and outcomes, respectively. We call $\mathbf{L}$ the loss matrix and $\mathbf{H}$ the feedback matrix. The matrix $\mathbf{L}$ is real-valued, while $\mathbf{H}$ is Σ-valued.

The problem of finite partial monitoring was introduced by Piccolboni and Schindelhauer [2001], who also introduced the algorithm FeedExp3. This algorithm differs from Exp3 only in how it estimates the losses $\hat{\ell}_i$. Denoting the loss and the feedback matrices as $\mathbf{L}$ and $\mathbf{H}$, and assuming that there exists a matrix K such that $\mathbf{L} = K\mathbf{H}$, the losses are estimated as $\hat{\ell}_i = K_{i,I_t} h_t / p_{I_t}$, where h_t is the feedback received. Again, it is not hard to see that, under the said condition, these estimated losses are unbiased estimates of the true unseen losses for all the actions.

Algorithm 6 The **FeedExp3** algorithm by Piccolboni and Schindelhauer [2001]

Input: $\mathbf{L}, \mathbf{H}, K$ matrices
Parameters: γ, η
Initialization: $w \leftarrow (1, \ldots, 1)$
for $t = 1, 2, \ldots$ **do**
  Play action I_t drawn from the multinomial distribution with parameters $p_i = (1-\gamma)w_i / \sum_{j=1}^{N} w_j + \gamma/N$, $i = 1, \ldots, N$
  Observe feedback $h_t = \mathbf{H}_{I_t, J_t}$
  for $i = 1, \ldots, N$ **do**
    Compute $\hat{\ell}_i \leftarrow K_{i,I_t} h_t / p_{I_t}$
    Update weights $w_i \leftarrow w_i e^{-\eta \hat{\ell}_i}$
  end for
end for

In their paper, Piccolboni and Schindelhauer [2001] prove an upper bound on the expected regret of $O(T^{3/4})$ for any learnable game. Later, this bound was strengthened to $O(T^{2/3})$ by Cesa-Bianchi et al. [2006], who also presented a specific partial monitoring game, a variant of the so-called label efficient prediction game, for which they prove a lower bound on the expected regret of $\Omega(T^{2/3})$. This result shows that the worst-case bound on the *class* of all non-trivial finite partial-monitoring games of $O(T^{2/3})$ is not improvable. However, as we see for example with multi-armed bandits, some non-trivial partial-monitoring games can have minimax regret growth rate better than $\Theta(T^{2/3})$. As cited from Cesa-Bianchi et al. [2006], *"it remains a challenging problem to characterize the class of problems that admit rates of convergence*[4] *faster than* $O(T^{-1/3})$*"*.

[4] In their paper they use the average per round regret instead of the cumulative regret.

Chapter 3

Summary of contributions

In this chapter, we list our contributions that will be described in details in the subsequent chapters.

3.1 Partial monitoring with two outcomes

We start with a characterization of the minimax regret (up to logarithmic factors) of almost all games with finitely many actions and *two* outcomes, against non-stochastic opponents (see Chapter 4). We show that, apart from a set of *degenerate* games (see Condition 2), partial-monitoring games can be categorized into four classes; *trivial* games with 0 minimax regret, *easy* games with $\widetilde{\Theta}(\sqrt{T})$ minimax regret, *hard* games with $\Theta(T^{2/3})$ minimax regret, and *hopeless* games with $\Theta(T)$ minimax regret. This classification result breaks down to the following theorems:

1. The minimax regret of a finite game is zero if and only if there exists an action that has always the smallest loss, independently of the outcome (Lemma 1).

2. All other finite games admit a lower bound for the minimax regret of $\Omega(\sqrt{T})$ (Theorem 4).

3. In a two-outcome game, if no action gives feedback information then the minimax regret is lower bounded by $\Omega(T)$.

4. If the condition in Case 3 does not apply, then the algorithm FeedExp3 by Piccolboni and Schindelhauer [2001] achieves $O(T^{2/3})$ minimax regret (proven by Cesa-Bianchi et al. [2006]).

5. For a two-outcome non-degenerate game, if the *separation condition* holds (see Definition 1), then a regret of $\widetilde{O}(\sqrt{T})$ is achievable. We introduce the algorithm AppleTree and prove the upper bound in Section 4.4.

6. For a two-outcome non-degenerate game, if the separation condition *does not* hold then the minimax regret is lower bounded by $\Omega(T^{2/3})$ (Theorem 5).

The six cases above give the characterization result.

3.2 Partial monitoring with two actions

Our next contribution deals with the "dual" case; after investigating games with two outcomes, we turn our attention to games with two actions (see Chapter 5). We show that if a game has only two actions then there are three categories: trivial and hopeless games with 0 and $\Theta(T)$ minimax regret, respectively, and a third category with $\Theta(\sqrt{T})$ minimax regret. This basically means that there are no "hard" games; any two-action game that is not trivial or hopeless is easy.

We prove the above result by showing that if a game is not trivial or hopeless, then with some trivial transformations, one can turn the game into a new one where the loss and feedback matrices are identical ($\mathbf{L} = \mathbf{H}$). This essentially means that, under the new game, the learner has bandit-like information in every round (see Theorem 6). Then, using an algorithm that works for bandit games, an $O(\sqrt{T})$ regret is achievable.

3.3 Classification of finite stochastic partial-monitoring games

We generalize our results on two-outcome and two-action games to games with any finite number of outcomes and actions, under the extra assumption that the opponent is stochastic. We show that the games can be categorized to the same four categories as two-outcome games; trivial, easy, hard, and hopeless games. The distinguishing condition between easy and hard games is the *local observability condition* (see Definition 10), a generalization of the separation condition. This condition ensures that actions that are in some sense neighbors (see Definition 8 for a precise description) can be used to estimate the difference of their expected losses.

We prove the upper bound of the minimax regret for easy games by introducing and analyzing the algorithm Balaton. The description of the algorithm can be found in Section 6.2.1, while the analysis is in Section 6.2.2. To complete the characterization result, we prove an $\Omega(T^{2/3})$ lower bound on the regret of games that do not satisfy the local observability condition (Theorem 11).

3.4 Better algorithms for finite stochastic partial monitoring

The algorithm Balaton was designed for the purpose of proving the classification result. In particular, it is shown that Balaton achieves $\widetilde{O}(\sqrt{T})$ regret on easy games. In Chapter 7 we turn our attention to designing algorithms with some improved properties.

First, we introduce the algorithm CBP-Vanilla. The advantages of CBP-Vanilla over Balaton are the following:

- It is an anytime algorithm: it does not need to know the time horizon (T) to achieve low regret.
- Apart from achieving near-optimal minimax regret (Corollary 1), it also achieves a logarithmic individual regret (Theorem 12).
- It performs significantly better empirically (see Section 7.2.3).

Then, we present the algorithm CBP, an extension of CBP-Vanilla, that is able to achieve near optimal minimax regret for both easy and hard games. The additional advantageous properties of CBP are:

- If run on an easy game, it simulates CBP-Vanilla.
- It achieves $O(T^{2/3})$ minimax regret for hard games (Corollary 2).
- If we appropriately restrict the space of strategies that the opponent can use, CBP achieves $\widetilde{O}(\sqrt{T})$ minimax regret.

The last assertion essentially means that if the opponent plays in an "easy region" of the game, then the game behaves the same way as if it was an easy game. For the precise statement, see Theorem 14.

We empirically compare two of our algorithms (CBP and Balaton) with FeedExp3 of Piccolboni and Schindelhauer [2001]. These empirical results are found in Section 7.2.3.

3.5 Online probing

In this work (see Chapter 8) we introduce a new online learning game where in every time step the learner has to predict a label based on some features received at the beginning of the turn. The learner has to decide which features to request, where each feature has an additional cost assigned to it. Additionally, there can be a cost of requesting the true label at the end of the turn. We define the regret as the cumulative loss of the learner compared to that of the best linear predictor.

We study two versions of the above game. In the first version, the cost of requesting the label is zero, therefore we can assume without loss of generality

that the learner asks for the true label in every time step. We show that the minimax regret of the game with free labels scales with the time horizon as $\widetilde{O}(\sqrt{T})$ (Theorem 15). In the second version, the cost of requesting the label is strictly positive. We show that in this case, the minimax regret scales as $\widetilde{\Theta}(T^{2/3})$ (Theorems 16 and 17). The difference between the regret growth rate of the two versions of the game is another nice indication of how the feedback structure can dramatically change the complexity of an online learning game.

Chapter 4

Two-outcome games[1]

The first games we investigate are partial-monitoring games with two outcomes and a finite number of actions. As we will see, these games serve as a stepping stone towards understanding games with any finite number of outcomes. In this chapter we show that, apart from a small set of *degenerate* games, partial-monitoring games with two outcomes fall into one of the following four categories:

1. *Trivial* games with minimax regret zero;
2. *Easy* games with minimax regret $\widetilde{\Theta}(\sqrt{T})$;
3. *Hard* games with minimax regret $\Theta(T^{2/3})$; and
4. *Hopeless* games with minimax regret $\Omega(T)$.

In particular, this classification result shows that there exist no non-degenerate games with minimax regret $\Theta(T^\alpha)$ for $1/2 < \alpha < 2/3$.

4.1 Basic definitions and notations

Remember that a finite partial-monitoring game $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ is specified by a pair of $N \times M$ matrices $(\mathbf{L}, \mathbf{H})$ where N is the number of actions, M is the number of outcomes, $\mathbf{L}$ is the *loss matrix*, and $\mathbf{H}$ is the *feedback matrix*. We use the notation $\underline{n} = \{1, \dots, n\}$ for any integer and denote the actions and outcomes by integers starting from 1, so the action set is $\underline{N}$ and the outcome set is $\underline{M}$. We denote by $\mathbf{L}[i, j]$ and $\mathbf{H}[i, j]$ ($i \in \underline{N}$, $j \in \underline{M}$) the entries of $\mathbf{L}$ and $\mathbf{H}$, respectively. We denote by ℓ_i the column vector consisting of the i^{th} row ($i \in \underline{N}$) of $\mathbf{L}$, and we call it the *loss vector of action* i. The elements of $\mathbf{L}$ are arbitrary real numbers. The elements of $\mathbf{H}$ belong to some alphabet Σ, we only assume that the learner is able to distinguish two different elements of the alphabet. We often use the set of natural or real numbers as the alphabet.

[1] Versions of the work in this chapter appeared in Bartók, Pál, and Szepesvári [2010] and Antos, Bartók, Pál, and Szepesvári [2012].

The matrices $\mathbf{L}$, $\mathbf{H}$ are known by both the learner and the opponent. The game proceeds in T rounds. In each round $t = 1, 2, \ldots, T$, the learner chooses an action $I_t \in \underline{N}$ and simultaneously the opponent chooses an outcome $J_t \in \underline{M}$. Next, the learner receives the feedback $\mathbf{H}[I_t, J_t]$. Nothing else is revealed to the learner; in particular J_t and the loss $\mathbf{L}[I_t, J_t]$ incurred by the learner remain hidden.

In this chapter we assume that the opponent is (oblivious) adversarial, that is, we assume that the sequence of outcomes $J_1, J_2, \ldots, J_T$ is a fixed deterministic sequence chosen before the first round of the game. A randomized strategy (algorithm) $\mathcal{A}$ of the learner is a sequence of random functions $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_T$ where each of the functions maps the feedback from the past outcomes (and learner's internal random "bits") to an action I_t; formally $\mathcal{A}_t : \Sigma^{t-1} \times \Omega \to \underline{N}$.

The goal of the learner is to keep his *cumulative loss* $\sum_{t=1}^{T} \mathbf{L}[I_t, J_t]$ small. With the notation of this chapter, the *(cumulative) regret* of an algorithm $\mathcal{A}$ is defined as

$$R_T = R_T^{\mathcal{A}}(\mathbf{G}) = \sum_{t=1}^{T} \mathbf{L}[I_t, J_t] - \min_{i \in \underline{N}} \sum_{t=1}^{T} \mathbf{L}[i, J_t] \ .$$

4.2 Characterization of games with two outcomes

In this section, we give the main characterization result of this chapter. We need a preliminary definition that is useful for any finite game:

Definition 3 (Properties of Actions)**.** *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be a finite partial-monitoring game with N actions and M outcomes. Let $i \in \underline{N}$ be one of its actions. Let Δ_M denote the set of M-dimensional probability vectors.*

- *Action i is called* dominated *if for any $p \in \Delta_M$ there exists an action i' such that $\ell_{i'} \neq \ell_i$ and $\ell_{i'}^\top p \leq \ell_i^\top p$.*
- *Action i is called* non-dominated *if it is not dominated.*
- *Action i is called* degenerate *if it is dominated and there exists a distribution $p \in \Delta_M$ such that for all $i' \in \underline{N}$, $\ell_i^\top p \leq \ell_{i'}^\top p$.*
- *Action i is called* all-revealing *if any pair of outcomes j, j', $j \neq j'$ satisfies $\mathbf{H}[i, j] \neq \mathbf{H}[i, j']$.*
- *Action i is called* none-revealing *if any pair of outcomes j,j' satisfies $\mathbf{H}[i, j] = \mathbf{H}[i, j']$.*
- *Action i is called* partially-revealing *if it is neither all-revealing, nor none-revealing.*

- *All-revealing and partially-revealing actions together are called* revealing *actions.*

- *Two or more actions with the same loss vector are called* duplicate *actions.*

The property of being dominated has an equivalent dual definition: action i is dominated if there exists a set of actions with their respective loss vectors different from ℓ_i such that some convex combination of their loss vectors componentwise lower bounds ℓ_i.

In games with $M = 2$ outcomes, each action is either all-revealing or none-revealing. This dichotomy is one of the key properties that lead to the classification theorem for two-outcome games. To emphasize the dichotomy, from now on we will refer to actions as *revealing* and *non-revealing* whenever it is clear from the context that $M = 2$.

This dichotomy also allows us to assume without of loss generality that there are no duplicate actions. Clearly, if multiple actions with the same loss vector exist, all but one can be removed (together with the corresponding rows of $\mathbf{L}$ and $\mathbf{H}$) without changing the minimax regret: If all of them are non-revealing, we keep one of the actions and remove all the others. Otherwise, we keep a revealing action and remove the others. Then replacing any algorithm by one that, instead of a removed action, chooses always the corresponding kept action, it is easy to see that the loss of the new algorithm cannot increase and equals the loss of this algorithm for the original game. Thus, the two games will have the same minimax regret.

The concepts of dominated and non-dominated actions can be visualized for two-outcome games by drawing the loss vector of each action as a point in $\mathbb{R}^2$. The points corresponding to the non-dominated actions lie on the bottom-left boundary of the convex hull of the set of all the actions, as shown in Figure 4.1. Enumerating the non-dominated actions ordered according to their loss for the first outcome gives rise to a sequence $(i_1, i_2, \ldots, i_K)$, which we call the *chain of non-dominated actions.*

To state the classification theorem, we introduce the following conditions.

Condition 1 (Separation condition)**.** *A two-outcome game* $\mathbf{G}$ *satisfies the* separation condition *if, after removing duplicate actions, its chain of non-dominated actions does* ***not*** *have a pair of consecutive actions* i_k*,* i_{k+1} *such that both of them are non-revealing. The set of games satisfying this condition will be denoted by* $\mathcal{S}$*.*

Condition 2 (Non-degeneracy condition)**.** *A two-outcome game* $\mathbf{G}$ *is* degenerate *if it has a degenerate revealing action. If* $\mathbf{G}$ *is not degenerate, we call it* non-degenerate *and we say that it satisfies the* non-degeneracy condition*.*

As we will soon see, the separation condition is the key to distinguish between *hard* and *easy* games. On the other hand, the non-degeneracy condition

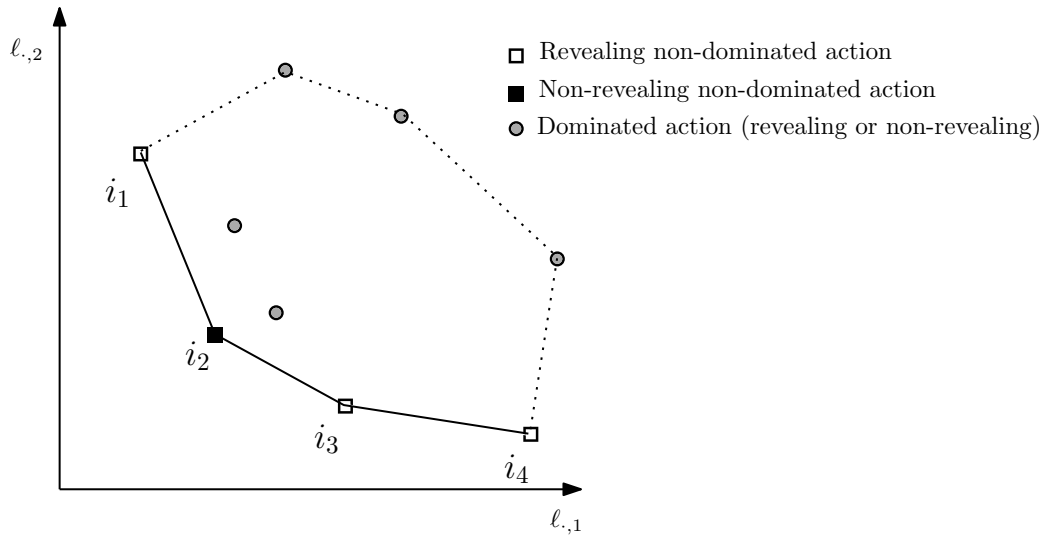


Figure 4.1: The figure shows each action i as a point in $\mathbb{R}^2$ with coordinates $(\mathbf{L}[i,1], \mathbf{L}[i,2])$. The solid line connects the chain of non-dominated actions, which, by convention are ordered according to their loss for the first outcome.

is merely a technical condition that we need in our proofs. The set of degenerate games is excluded from the characterization (in Chapter 6 this gap will be filled in). With this preparations, we are now ready to state our main result.

Theorem 2 (Classification of Two-Outcome Partial-Monitoring Games)**.** *Let $\mathcal{S}$ be the set of all finite partial-monitoring games with two outcomes that satisfy the separation condition. Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be a game with two outcomes that satisfies the non-degeneracy condition. Let K be the number of non-dominated actions in $\mathbf{G}$, counting duplicate actions only once. The minimax expected regret $\mathrm{R}_T(\mathbf{G})$ satisfies*

$$\mathrm{R}_T(\mathbf{G}) = \begin{cases} 0, & K = 1; \quad (4.1a) \\ \widetilde{\Theta}\left(\sqrt{T}\right), & K \geq 2,\ \mathbf{G} \in \mathcal{S}; \quad (4.1b) \\ \Theta\left(T^{2/3}\right), & K \geq 2,\ \mathbf{G} \notin \mathcal{S},\ \mathbf{G} \text{ has a revealing action}; \quad (4.1c) \\ \Theta(T), & \text{otherwise.} \quad (4.1d) \end{cases}$$

We call the games in cases (4.1a)–(4.1d) *trivial*, *easy*, *hard*, and *hopeless*, respectively. Case (4.1a) is proven by the following lemma which shows that a trivial game is also characterized by having 0 minimax regret in a single round or by having an action "dominating" alone all the others:

Lemma 1. *For any finite partial-monitoring game, the following four statements are equivalent:*

a) The minimax regret is zero for each T.

b) The minimax regret is zero for some T.

c) There exists a (non-dominated) action $i \in \underline{N}$ whose loss is not larger than the loss of any other action irrespectively of the choice of opponent's action.

d) The number of non-dominated actions is one ($K = 1$).

The proof of Lemma 1, as all the other lemmas of this thesis, can be found in the Appendix.

Case (4.1d) of Theorem 2 is proven the following way[2]:

Proof of Theorem 2 Case (4.1d). We know that $K \geq 2$ and $\mathbf{G}$ has no revealing action. Then for any algorithm $\mathcal{A}$,

$$\begin{aligned}
\mathbb{E}[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] &\geq \sup_{j \in \underline{M}, J_1 = \cdots = J_T = j} \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{L}[I_t, J_t] - \min_{i \in \underline{N}} \sum_{t=1}^{T} \mathbf{L}[i, J_t]\right] \\
&\geq \frac{1}{M} \sum_{j=1}^{M} \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{L}[I_t, j] - T \min_{i \in \underline{N}} \mathbf{L}[i, j]\right] \\
&= \frac{1}{M} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{j=1}^{M} \mathbf{L}[I_t, j]\right] - \frac{T}{M} \sum_{j=1}^{M} \min_{i \in \underline{N}} \mathbf{L}[i, j] \; .
\end{aligned}$$

Here I_t is a random variable usually depending on $J_{1:T-1}$, that is, on j through the outcomes. However, since $\mathbf{G}$ has no revealing action, now the distribution of I_t is independent of j, thus $\mathbb{E}[\sum_{j=1}^{M} \mathbf{L}[I_t, j]] \geq \min_{i \in \underline{N}} \sum_{j=1}^{M} \mathbf{L}[i, j]$ for each t, and we have

$$\mathbb{E}[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] \geq T \underbrace{\frac{1}{M}\left[\min_{i \in \underline{N}} \sum_{j=1}^{M} \mathbf{L}[i, j] - \sum_{j=1}^{M} \min_{i \in \underline{N}} \mathbf{L}[i, j]\right]}_{c} = cT \; ,$$

where $c > 0$ if $K \geq 2$ (because $c \geq 0$, and $c = 0$ would imply Lemma 1 c), thus also d)). Since c depends only on $\mathbf{L}$, $\mathbb{E}[\mathrm{R}_T(\mathbf{G})] \geq cT = \Theta(T)$. □

The upper bound of case (4.1c) can be derived from a result of Cesa-Bianchi and Lugosi [2006]: Note that the entries of $\mathbf{H}$ can be changed without changing the information revealed to the learner as long as one does not change the pattern of which elements in a row are equal and different. Cesa-Bianchi and Lugosi [2006, Theorem 6.5] show that if the entries of $\mathbf{H}$ can be chosen such that

$$\operatorname{rank}(\mathbf{H}) = \operatorname{rank}\left(\begin{pmatrix}\mathbf{H} \\ \mathbf{L}\end{pmatrix}\right)$$

then $O(T^{2/3})$ expected regret is achievable. This condition holds trivially for two-outcome games with at least one revealing action and $N \geq 2$. It remains to prove the upper bound for case (4.1b), the lower bound for (4.1b), and the lower bound for (4.1c); we prove these in Sections 4.4, 4.5, and 4.6, respectively.

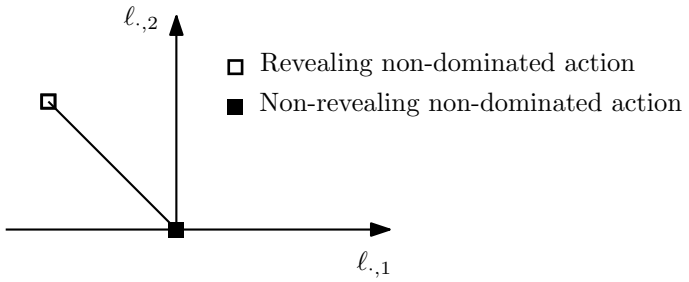
[2]Note that the linear lower bound could also be derived from the result of Piccolboni and Schindelhauer [2001] but in the case of two outcomes, it is worthwile to show that there is a much simpler proof.

4.3 Examples

Before we dive into the proof of the remaining parts of Theorem 2, we give a few examples of finite partial-monitoring games with two outcomes and show how the theorem can be applied. For each example we present the matrices $\mathbf{L}, \mathbf{H}$ and depict the loss vectors of actions as points in $\mathbb{R}^2$.

Example 7. [One-Armed Bandit] We start with an example of a multi-armed bandit game. Multi-armed bandit games are those where the feedback equals the instantaneous loss, that is, when $\mathbf{L} = \mathbf{H}$.[3]

$$\mathbf{L} = \begin{pmatrix} 0 & 0 \\ -1 & 1 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 0 & 0 \\ -1 & 1 \end{pmatrix}.$$

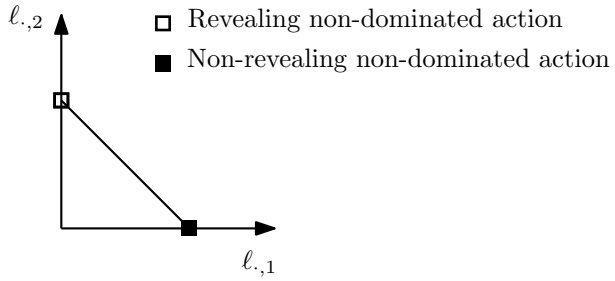


Because the loss of the first action is 0 regardless of the outcome, and the loss varies only for the second action, we call this game a *one-armed bandit* game. Both actions are non-dominated and the second one is revealing, therefore this is an easy game and according to Theorem 2 the minimax regret is $\widetilde{\Theta}(\sqrt{T})$. (For this specific game, it can be shown that the minimax regret is in fact $\Theta(\sqrt{T})$.)

Example 8. [Apple Tasting] Consider an orchard that wants to hand out its crop of apples for sale. However, some of the apples might be rotten. The orchard can do a sequential test. Each apple can be either tasted (which reveals whether the apple is healthy or rotten) or the apple can be given out for sale. If a rotten apple is given out for sale, the orchard suffers a unit loss. On the other hand, if a healthy apple is tasted, it cannot be sold and, again, the orchard suffers a unit loss. This can be formalized by the following partial-monitoring game [Helmbold et al., 2000]:

[3] "Classically", non-stochastic multi-armed bandit problems are defined by the restriction that in no round can the learner gain any information about the losses of actions other than the chosen one, that is, $\mathbf{L}$ is not known in advance to the learner. (Also, the domain set of losses is often infinite there ($M = \infty$).) When $\mathbf{H} = \mathbf{L}$ in our setting, depending on $\mathbf{L}$, this might or might not be the case; the "classical bandit" problem with losses constrained to a finite set is a special case of games with $\mathbf{H} = \mathbf{L}$. However, the latter condition also allows other types of games where the learner can recover the losses of actions not chosen, and so which could be "easier" than classical bandits due to the knowledge of $\mathbf{L}$. Nevertheless, it is easy to see that these games are *at most* as hard as classical bandit games.

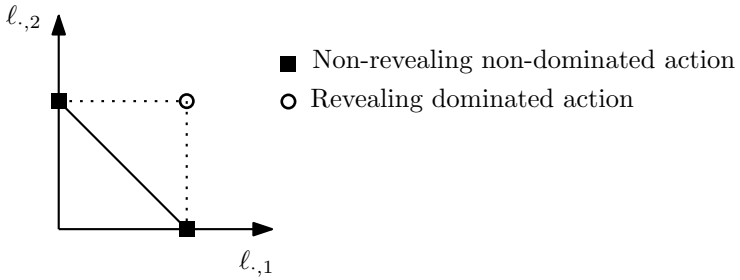
$$\mathbf{L} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} a & a \\ b & c \end{pmatrix}.$$



Here, the set of feedbacks has three elements: $\Sigma = \{a, b, c\}$. The first action corresponds to giving out the apple for sale, the second corresponds to tasting the apple; the first outcome corresponds to a rotten apple, the second outcome corresponds to a healthy apple. Both actions are non-dominated and the second one is revealing, therefore this is an easy game and according to Theorem 2 the minimax regret is $\widetilde{\Theta}(\sqrt{T})$. This is apparently a new result for this game. Also notice that the picture is just a translation of the picture for the one-armed bandit.

Example 9. [Label Efficient Prediction] Consider a situation when we would like to sequentially classify emails as spam or as legitimate. For each email we have to output a prediction, and additionally we can request, as feedback, the correct label from the user. If we classify an email incorrectly or we request its label, we suffer a unit loss. (If the email is classified correctly and we do not request the feedback, no loss is suffered.) This can be formalized by the following partial-monitoring game [Cesa-Bianchi and Lugosi, 2006]:

$$\mathbf{L} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} a & b \\ c & c \\ d & d \end{pmatrix}.$$



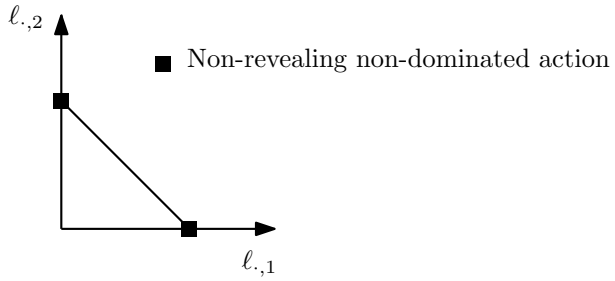
Here, the set of feedbacks has four elements: $\Sigma = \{a, b, c, d\}$, the first action corresponds to a label request, and the second and the third action correspond

to a prediction (spam and legitimate, respectively) without a request. The outcomes correspond to spam and legitimate emails.

We see that the chain of non-dominated actions contains two neighboring non-revealing actions and there is a dominated revealing action. Therefore, this is a hard game and, by Theorem 2, the minimax regret is $\Theta(T^{2/3})$. This specific example was the only non-trivial game known before our work with minimax regret at least $\Omega(T^{2/3})$ [Cesa-Bianchi et al., 2006, Theorem 5.1].

Example 10. [A Hopeless Game] The following game is an example where the feedback does not reveal any information about the outcome:

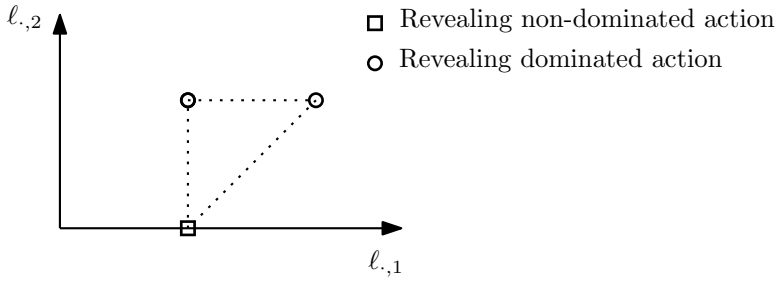
$$\mathbf{L} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} a & a \\ b & b \end{pmatrix}.$$



Here, the set of feedbacks has two elements: $\Sigma = \{a, b\}$. Because both actions are non-revealing and non-dominated, this is a hopeless game and thus its minimax regret is $\Theta(T)$.

Example 11. [A Trivial Game] In the following game, the best action, regardless of the outcome sequence, is action 2. A learner that chooses this action in every round is guaranteed to have zero regret.

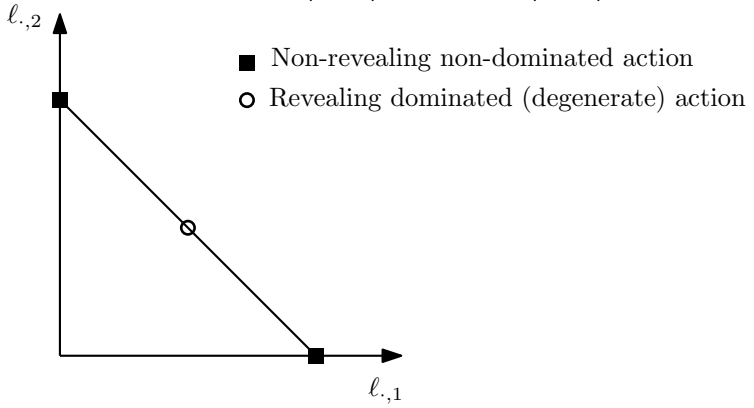
$$\mathbf{L} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix}.$$



Here, the set of feedbacks has six elements: $\Sigma = \{a, b, c, d, e, f\}$. Because this game has only one non-dominated action (action 2), it is a trivial game and thus its minimax regret is 0.

Example 12. [A Degenerate Game] The next game does not satisfy the non-degeneracy condition and therefore Theorem 2 does not apply.

$$\mathbf{L} = \begin{pmatrix} 2 & 0 \\ 1 & 1 \\ 0 & 2 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} a & a \\ b & c \\ d & d \end{pmatrix}$$



Here, the set of feedbacks has four elements: $\Sigma = \{a, b, c, d\}$. The minimax regret of this game is between $\Omega(\sqrt{T})$ and $O(T^{2/3})$. It remains an open problem to close this gap and determine the exact rate of growth.

4.4 Upper bound for easy games

In this section we present our algorithm for games satisfying the separation condition and the non-degeneracy condition, and prove that this algorithm achieves $\widetilde{O}(\sqrt{T})$ regret with high probability. We call the algorithm Apple-Tree since it builds a binary tree, leaves of which are apple tasting games.

4.4.1 The algorithm

In the first step of the algorithm we can simplify the game by first removing the dominated actions and then the duplicates as mentioned beforehand.

The idea of the algorithm is to recursively split the game until we arrive at games with two actions only. Now, if one has only two actions in a partial-information game, the game must be either a full-information game (if both actions are revealing) or an instance of a one-armed bandit (with one revealing and one non-revealing action).

To see why this latter case corresponds to one-armed bandits, assume without loss of generality that the first action is the revealing action. Now, it is easy to see that the regret of a sequence of actions in a game does not change if the loss matrix is changed by subtracting the same number from a column.[4]

[4]As a result, for any algorithm, if R_T is its regret at time T when measured in the game with the modified loss matrix, the algorithm's "true" regret will also be R_T (*i.e.,*

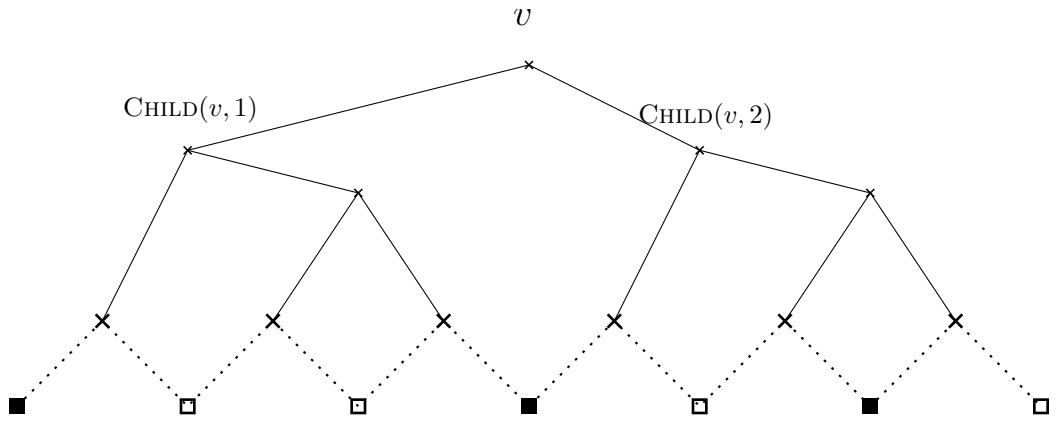


Figure 4.2: The binary tree built by the algorithm. The leaf nodes represent neighboring action pairs.

By subtracting $\mathbf{L}[2,1]$ from the first and $\mathbf{L}[2,2]$ from the second column we thus get the "equivalent" game where the second row of the loss matrix is zero, arriving at a one-armed bandit game (see Example 7). Since a one-armed bandit is a special form of a two-armed bandit, one can use Exp3.P due to Auer et al. [2003] to achieve the $O(\sqrt{T})$ regret.

Now, if there are more than two actions in the game, then the game is split, putting the first half of the actions into the first and the second half into the second subgame, with a *single common shared action*. Recall that, in the chain of non-dominated actions, the actions are ordered according to their losses corresponding to the *first* outcome. This is continued until the split results in games with two actions only. The recursive splitting of the game results in a binary tree (see Figure 4.2). The idea of the strategy played at an internal node of the tree is as follows: An outcome sequence of length T determines the frequency ρ_T of outcome 2. If this frequency is small, the optimal action is one of the actions of $\mathbf{G}_1$, the first subgame (simply because then the frequency of outcome 1 is high and $\mathbf{G}_1$ contains the actions with the smallest loss for the first outcome). Conversely, if this frequency is large, the optimal action is one of the actions of $\mathbf{G}_2$. In some intermediate range, the optimal action is the action shared between the subgames. Let the boundaries of this range be $\rho_1^* < \rho_2^*$ (ρ_1^* is thus the solution to $(1-\rho)\mathbf{L}[s-1,1]+\rho\mathbf{L}[s-1,2] = (1-\rho)\mathbf{L}[s,1]+\rho\mathbf{L}[s,2]$ and ρ_2^* is the solution to $(1-\rho)\mathbf{L}[s+1,1]+\rho\mathbf{L}[s+1,2] = (1-\rho)\mathbf{L}[s,1]+\rho\mathbf{L}[s,2]$, where $s = \lceil K/2 \rceil$ is the index of the action shared between the two subgames.)

If we knew ρ_T, a good solution would be to play a strategy where the actions are restricted to that of either game $\mathbf{G}_1$ or $\mathbf{G}_2$, depending on whether $\rho_T \le \rho_1^*$ or $\rho_T \ge \rho_2^*$. (When $\rho_1^* \le \rho_T \le \rho_2^*$ then it does not matter which action-set we restrict the play to, since the optimal action in this case is included in both sets.) There are two difficulties. First, since the outcome sequence is not known in advance, the best we can hope for is to know the

the algorithm's regret when measured in the original, unmodified game). Piccolboni and Schindelhauer [2001] exploit this idea, too.

running frequencies $\rho_t = \frac{1}{t}\sum_{s=1}^{t} \mathbb{I}_{\{J_s=2\}}$. However, since the game is a partial-information game, the outcomes are not revealed in all time steps, hence, even ρ_t is inaccessible. Nevertheless, for now let us assume that ρ_t was available. Then one idea would be to play a strategy restricted to the actions of either game $\mathbf{G}_1$ or $\mathbf{G}_2$ as long as ρ_t stays below ρ_1^* or above ρ_2^*. Further, when ρ_t becomes larger than ρ_2^* while previously the strategy played the action of $\mathbf{G}_1$ then we have to switch to the game $\mathbf{G}_2$. In this case, we start a fresh copy (a *reset*) of a strategy playing in $\mathbf{G}_2$. The same happens when a switch from $\mathbf{G}_2$ to game $\mathbf{G}_1$ is necessary. These resets are necessary because at the leaves we play according to strategies that use weights that depend on the cumulated losses of the actions *exponentially*. To see an example when without resets the algorithm fails to achieve a small regret, consider the case when there are 3 actions, the middle one being revealing. Assume that during the first $T/2$ time steps the frequency of outcome 2 oscillates between the two boundaries so that the algorithm switches constantly back and forth between the games $\mathbf{G}_1$ and $\mathbf{G}_2$. Assume further that in the second half of the game, the outcome is always 2. This way the optimal action will be 3. Nevertheless, up to time step $T/2$, the player of $\mathbf{G}_2$ will only see outcome 1 and thus will think that action 2 is the optimal action. In the second half of the game, he will not have enough time to recover and will play action 2 for too long. Resetting the algorithms of the subgames avoids this behavior.

If the number of switches was large, the repeated resetting of the strategies could be equally problematic. Luckily this cannot happen, hence the resetting does minimal harm. We will in fact show that this generalizes to the case even when ρ_t is estimated based on partial feedback (see Lemma 2).

Let us now turn to how ρ_t is estimated. As mentioned in Section 4.2, mapping a row of $\mathbf{H}$ bijectively leads to an equivalent game, thus for $M = 2$ we can assume without loss of generality that in any round, the algorithm receives (possibly random) feedback $H_t \in \{1, 2, *\}$: if a revealing action is played in the round, $H_t = J_t \in \{1, 2\}$, otherwise $H_t = *$. Let $\mathcal{H}_{1:t-1} = (I_1, H_1, \ldots, I_{t-1}, H_{t-1}) \in (\underline{N} \times \Sigma)^{t-1}$, the (random) history of actions and observations up to time step $t-1$. If the algorithm choosing the actions decides with probability $p_t \in (0, 1]$ to play a revealing action (p_t can depend on $\mathcal{H}_{1:t-1}$) then $\mathbb{I}_{\{H_t=2\}}/p_t$ is a simple unbiased estimate of $\mathbb{I}_{\{J_t=2\}}$ (in fact, $\mathbb{E}\left[\mathbb{I}_{\{H_t=2\}}/p_t | \mathcal{H}_{1:t-1}\right] = \mathbb{I}_{\{J_t=2\}}$). As long as p_t does not drop to a too low value, $\hat{\rho}_t = \frac{1}{t}\sum_{s=1}^{t} \frac{\mathbb{I}_{\{H_s=2\}}}{p_s}$ will be a relatively reliable estimate of ρ_t (see Lemma 3). However reliable this estimate is, it can still differ from ρ_t. For this reason, we push the boundaries determining game switches towards each other:

$$\rho_1' = \frac{2\rho_1^* + \rho_2^*}{3}, \quad \rho_2' = \frac{\rho_1^* + 2\rho_2^*}{3}. \tag{4.2}$$

We call the resulting algorithm AppleTree, because the elementary partial-information 2-action games in the bottom essentially correspond to instances of the apple tasting problem (see Example 8). The algorithm's main entry point is shown on Algorithm 7. Its inputs are the game $\mathbf{G} = (\mathbf{L}, \mathbf{H})$, the

Algorithm 7 The entry point of the AppleTree algorithm Main($\mathbf{G}, T, \delta$)

Input: $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ is a game, T is a horizon, $0 < \delta < 1$ is a confidence parameter
$\mathbf{G} \leftarrow$ Purify($\mathbf{G}$)
BuildTree(**root**, $\mathbf{G}, \delta$)
for $t \leftarrow 1$ **to** T **do**
  Play(**root**)
end for

Algorithm 8 The initialization routine InitEta($\mathbf{G}, T$).

Input: $\mathbf{G}$ is a game, T is a horizon
if IsRevealing($\mathbf{G}, 2$) **then**
  $\eta(v) \leftarrow \sqrt{8 \ln 2 \,/ T}$
else
  $\eta(v) \leftarrow \gamma(v)/4$
end if

time horizon and a confidence parameter $0 < \delta < 1$. The algorithm first eliminates the dominated and duplicate actions. This is followed by building a tree that is used to store variables necessary to play in the subgames (Algorithm 9): If the number of actions is 2, the procedure initializes various parameters that are used either by a bandit algorithm (based on Exp3.P [Auer et al., 2003]), or by the Weighted Majority algorithm (WM) by Littlestone and Warmuth [1994] (see Algorithm in Chapter 2). In the other case, it calls itself recursively on the split subgames and with an appropriately decreased confidence parameter.

The main worker routine is called Play. This is again a recursive function (see Algorithm 10). The special case when the number of actions is two is handled in routine PlayAtLeaf, which will be discussed later. When the number of actions is larger, the algorithm recurses to play in the subgame that was remembered as the game to be preferred from the last round and then updates its estimate of the frequency of outcome 2 based on the information received. When this estimate changes so that a switch of the current preferred game is necessary, the algorithm resets the algorithms in the subtree corresponding to the game switched to, and changes the variable storing the index of the preferred game. The Reset function used for this purpose, shown on Algorithm 11, is also recursive.

At the leaves, when there are only two actions, either WM or Exp3.P is used. These algorithms are used with their standard optimized parameters (see Corollary 4.2 for the tuning of WM, and Theorem 6.10 for the tuning of Exp3.P, both from the book of Cesa-Bianchi and Lugosi [2006]). For completeness, their pseudocodes are shown in Algorithms 12–13. Note that with Exp3.P we use the loss matrix transformation described earlier, hence the loss matrix has zero entries for the second (non-revealing) action, while the entry

Algorithm 9 The tree building procedure BUILDTREE$(v, \mathbf{G}, \delta)$

Input: $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ is a game, v is a tree node
if NUMOFACTIONS$(\mathbf{G})$= 2 **then**
  if not ISREVEALING$(\mathbf{G}, 1)$ **then**
    $\mathbf{G} \leftarrow$ SWAPACTIONS$(\mathbf{G})$
  end if
  $w_i(v) \leftarrow 1/2,\ i = 1, 2$
  $\beta(v) \leftarrow \sqrt{\ln(2/\delta)/(2T)}$
  $\gamma(v) \leftarrow 8\beta(v)/(3 + \beta(v))$
  INITETA$(\mathbf{G}, T)$
else
  $(\mathbf{G}_1, \mathbf{G}_2) \leftarrow$ SPLITGAME$(\mathbf{G})$
  BUILDTREE(CHILD$(v, 1)$, $\mathbf{G}_1, \delta/(4T)$)
  BUILDTREE(CHILD$(v, 2)$, $\mathbf{G}_2, \delta/(4T)$)
  $g(v) \leftarrow 1$, $\hat{\rho}(v) \leftarrow 0$, $t(v) \leftarrow 1$
  $(\rho_1'(v), \rho_2'(v)) \leftarrow$ BOUNDARIES$(\mathbf{G})$
end if
$\mathbf{G}(v) \leftarrow \mathbf{G}$

for action 1 and outcome j is $\mathbf{L}[1, j](v) - \mathbf{L}[2, j](v)$. Here $\mathbf{L}[i, j](v)$ stands for the loss of action i and outcome j in the game $\mathbf{G}(v)$ that is stored at node v.

4.4.2 Proof of the upper bound

Theorem 3. *Assume* $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ *satisfies the separation condition and the non-degeneracy condition and* $\mathbf{L}[i, j] \leq 1$. *Denote by* $\widehat{\mathrm{R}}_T$ *the regret of Algorithm* APPLETREE *up to time step* T. *There exist constants c,p such that for any* $0 < \delta < 1$ *and* $T \in \mathbb{N}$, *for any outcome sequence* $J_1, \ldots, J_T$, *the algorithm with input* $\mathbf{G}, T, \delta$ *achieves* $\mathbb{P}\left(\widehat{\mathrm{R}}_T \leq c\sqrt{T}\ln^p(2T/\delta)\right) \geq 1 - \delta$.

Throughout the proof we will analyze the algorithm's behavior at the root node. We will use time indices as follows. Let us define the filtration $\{\mathcal{F}_t = \sigma(I_1, \ldots, I_t)\}_t$, where I_t is the action the algorithm plays at time step t. For any variable $x(v)$ used by the algorithm, we will use $x_t(v)$ to denote the value of $x(v)$ that is measurable with respect to $\mathcal{F}_t$, but not measurable with respect to $\mathcal{F}_{t-1}$. From now on, we also abbreviate $x_t(\text{root})$ by x_t. We start with two lemmas. The first lemma shows that the number of switches the algorithm makes is small.

Lemma 2. *Let* S *be the number of times* APPLETREE *calls* RESET *at the root node. Then there exists a universal constant* c^* *such that* $S \leq \frac{c^* \ln T}{\Delta}$, *where* $\Delta = \rho_2' - \rho_1'$ *with* ρ_1' *and* ρ_2' *given by* (4.2).

Note that here we use the non-degeneracy condition to ensure that $\Delta > 0$.

Algorithm 10 The recursive function PLAY(v)

Input: v is a tree node
if NUMOFACTIONS($\mathbf{G}(v)$)$= 2$ **then**
  $(p, h) \leftarrow$ PLAYATLEAF(v)
else
  $(p, h) \leftarrow$ PLAY(CHILD($v, g(v)$))
  $\hat{\rho}(v) \leftarrow (1 - \frac{1}{t(v)})\hat{\rho}(v) + \frac{1}{t(v)} \frac{\mathbb{I}_{\{h=2\}}}{p}$
  if $g(v) = 2$ **and** $\hat{\rho}(v) < \rho_1'(v)$ **then**
    RESET(CHILD($v, 1$)); $g(v) \leftarrow 1$
  else if $g(v) = 1$ **and** $\hat{\rho}(v) > \rho_2'(v)$ **then**
    RESET(CHILD($v, 2$)); $g(v) \leftarrow 2$
  end if
  $t(v) \leftarrow t(v) + 1$
end if
Return (p, h)

Algorithm 11 Function RESET(v)

Input: v is a tree node
if NUMOFACTIONS($\mathbf{G}(v)$)$= 2$ **then**
  $w_i(v) \leftarrow 1/2$, $i \leftarrow 1, 2$
else
  $g(v) \leftarrow 1$, $\hat{\rho}(v) \leftarrow 0$, $t(v) \leftarrow 1$
  RESET(CHILD($v, 1$))
end if

The next lemma shows that the estimate of the relative frequency of outcome 2 is not far away from its true value.

Lemma 3. *For any $0 < \delta < 1$, with probability at least $1 - \delta$, for all $t \geq 8\sqrt{T}\ln(2T/\delta)/(3\Delta^2)$, $|\hat{\rho}_t - \rho_t| \leq \Delta$.*

Proof of Theorem 3. To prove that the algorithm achieves the desired regret bound we use induction on the depth of the tree, d. If $d = 1$, APPLETREE plays either WM or Exp3.P. WM is known to satisfy Theorem 3, and, as we discussed earlier, Exp3.P achieves $O(\sqrt{T}\ln T/\delta)$ regret as well. As the induction hypothesis we assume that Theorem 3 is true for any T and any game such that the tree built by the algorithm has depth $d' < d$.

Let $Q_1 = \{1, \ldots, \lceil K/2 \rceil\}$, $Q_2 = \{\lceil K/2 \rceil, \ldots, K\}$ be the sets of actions associated with the subgames in the root. (Recall that the actions are ordered with respect to $\mathbf{L}[\cdot, 1]$.) Furthermore, let us define the following values: Let $T_0^0 = 1$, let T_i^0 be the first time step t after T_{i-1}^0 such that $g_t \neq g_{t-1}$. In other words, T_i^0 are the time steps when the algorithm switches between the subgames. Finally, let $T_i = \min(T_i^0, T + 1)$. From Lemma 2 we know that $T_{S_{\max}+1} = T + 1$, where $S_{\max} = \frac{c^* \ln T}{\Delta}$. It is easy to see that T_i are stopping times for any $i \geq 1$.

Algorithm 12 Function PLAYATLEAF(v)

Input: v is a tree node
if REVEALINGACTIONNUMBER($\mathbf{G}(v)$)$= 2$ **then**
  {Full-information case}
  $(p, h) \leftarrow \mathrm{WM}(v)$
else
  $p \leftarrow (1-\gamma(v))\frac{w_1(v)}{w_1(v)+w_2(v)} + \gamma(v)/2$
  $U \sim \mathcal{U}_{[0,1)}$ {U is uniform in $[0,1)$}
  if $U < p$ **then**
    $h \leftarrow \mathrm{CHOOSE}(1)$ {$h \in \{1,2\}$}
    $L_1 \leftarrow (\mathbf{L}[1,h](v) - \mathbf{L}[2,h](v) + \beta(v))/p$
    $L_2 \leftarrow \beta(v)/(1-p)$
    $w_1(v) \leftarrow w_1(v)\exp(-\eta(v)L_1)$
    $w_2(v) \leftarrow w_2(v)\exp(-\eta(v)L_2)$
  else
    $h \leftarrow \mathrm{CHOOSE}(2)$ {here $h = *$}
  end if
end if
Return(p, h)

Without loss of generality, from now on we will assume that the optimal action $i^* \in Q_1$. If $i^* = \lceil K/2 \rceil$ then, since it is contained in both subgames, the bound trivially follows from the induction hypothesis and Lemma 2. In the rest of the proof we assume $i^* < K/2$.

Let $S = \max\{i \geq 1 \mid T_i^0 \leq T\}$ be the number of switches, $c = \frac{8}{3\Delta^2}$, and $\mathcal{B}$ be the event that for all $t \geq c\sqrt{T}\ln(4T/\delta)$, $|\hat{\rho}_t - \rho_t| \leq \Delta$. We know from Lemma 3 that $\mathbb{P}(\mathcal{B}) \geq 1 - \delta/2$. On $\mathcal{B}$ we have that $|\hat{\rho}_T - \rho_T| \leq \Delta$, and thus, using that $i^* < K/2$, $\rho_T \leq \rho_1^*$. This implies that in the last phase the algorithm plays on $\mathbf{G}_1$. It is also easy to see that before the last switch, at time step $T_S - 1$, $\hat{\rho}$ is between ρ_1^* and ρ_2^*, if T_S is large enough. Thus, up to time step $T_S - 1$, the optimal action is $\lceil K/2 \rceil$, the one that is shared by the two subgames. This implies that $\sum_{t=1}^{T_S-1} \mathbf{L}[i^*, J_t] - \mathbf{L}[\lceil K/2 \rceil, J_t] \geq 0$. On the other hand, if $T_S \leq c\sqrt{T}\ln(4T/\delta)$ then

$$\sum_{t=1}^{T_S-1} \mathbf{L}[i^*, J_t] - \mathbf{L}[\lceil K/2 \rceil, J_t] \geq -c\sqrt{T}\ln(4T/\delta) \ .$$

Algorithm 13 Function WM(v)

Input: v is a tree node
$p \leftarrow \frac{w_1(v)}{w_1(v)+w_2(v)}$
$U \sim \mathcal{U}_{[0,1)}$ $\{U$ is uniform in $[0,1)\}$
if $U < p$ **then**
  $I \leftarrow 1$
else
  $I \leftarrow 2$
end if
$h \leftarrow \text{CHOOSE}(I)$ $\{h \in \{1,2\}\}$
$w_1(v) \leftarrow w_1(v)\exp(-\eta(v)\mathbf{L}[1,h](v))$
$w_2(v) \leftarrow w_2(v)\exp(-\eta(v)\mathbf{L}[2,h](v))$
Return(p,h)

Thus, we have

$$\begin{aligned}
\widehat{\mathrm{R}}_T &= \sum_{t=1}^{T} \mathbf{L}[I_t, J_t] - \mathbf{L}[i^*, J_t] \\
&= \sum_{t=1}^{T_S-1} \left(\mathbf{L}[I_t, J_t] - \mathbf{L}[i^*, J_t]\right) + \sum_{t=T_S}^{T} \left(\mathbf{L}[I_t, J_t] - \mathbf{L}[i^*, J_t]\right) \\
&\le \mathbb{I}_{\{\mathcal{B}\}} \left(\sum_{t=1}^{T_S-1} \left(\mathbf{L}[I_t, J_t] - \mathbf{L}[\lceil K/2 \rceil, J_t]\right) + \sum_{t=T_S}^{T} \left(\mathbf{L}[I_t, J_t] - \mathbf{L}[i^*, J_t]\right) \right) \\
&\quad + \underbrace{c\sqrt{T}\ln(4T/\delta) + \left(\mathbb{I}_{\{\mathcal{B}^c\}}\right) T}_{D} \\
&\le D + \mathbb{I}_{\{\mathcal{B}\}} \sum_{r=1}^{S_{\max}} \max_{i \in Q_{\pi(r)}} \sum_{t=T_{r-1}}^{T_r-1} \left(\mathbf{L}[I_t, J_t] - \mathbf{L}[i, J_t]\right) \\
&= D + \mathbb{I}_{\{\mathcal{B}\}} \sum_{r=1}^{S_{\max}} \max_{i \in Q_{\pi(r)}} \sum_{m=1}^{T} \mathbb{I}_{\{T_r - T_{r-1} = m\}} \sum_{t=T_{r-1}}^{T_{r-1}+m-1} \left(\mathbf{L}[I_t, J_t] - \mathbf{L}[i, J_t]\right) \ ,
\end{aligned}$$

where $\pi(r)$ is 1 if r is odd and 2 if r is even. Note that for the last line of the above inequality chain to be well defined, we need outcome sequences of length at most $2T$. It does us no harm to assume that for all $T < t \le 2T$, say, $J_t = 1$.

Recall that the strategies that play in the subgames are reset after the switches. Hence, the sum $\widehat{\mathrm{R}}_m^{(r)} = \sum_{t=T_{r-1}}^{T_{r-1}+m-1} \left(\mathbf{L}[I_t, J_t] - \mathbf{L}[i, J_t]\right)$ is the regret of the algorithm if it is used in the subgame $\mathbf{G}_{\pi(r)}$ for $m \le T$ steps. Then, exploiting that T_r are stopping times, we can use the induction hypothesis to bound $\widehat{\mathrm{R}}_m^{(r)}$. In particular, let $\mathcal{C}$ be the event that for all $m \le T$ the sum is less than $c\sqrt{T}\ln^p(2T^2/\delta)$. Since the root node calls its children with confidence

parameter $\delta/(2T)$, we have that $\mathbb{P}\left(\mathcal{C}^c\right) \leq \delta/2$. In summary,

$$\begin{aligned}\widehat{\mathrm{R}}_T &\leq D + \mathbb{I}_{\{\mathcal{C}^c\}} T + \mathbb{I}_{\{\mathcal{B}\}}\mathbb{I}_{\{\mathcal{C}\}} S_{\max} c\sqrt{T}\ln^p 2T^2/\delta \\ &\leq \mathbb{I}_{\{\mathcal{B}^c\cup\mathcal{C}^c\}} T + c\sqrt{T}\ln(4T/\delta) + \mathbb{I}_{\{\mathcal{B}\}}\mathbb{I}_{\{\mathcal{C}\}}\frac{c^*\ln T}{\Delta} c\sqrt{T}\ln^p 2T^2/\delta.\end{aligned}$$

Thus, on $\mathcal{B}\cap\mathcal{C}$, $\widehat{\mathrm{R}}_T \leq \frac{2^p c c^*}{\Delta}\sqrt{T}\ln^{p+1}\left(2T/\delta\right)$, which, together with $\mathbb{P}\left(\mathcal{B}^c \cup \mathcal{C}^c\right) \leq \delta$ concludes the proof. □

Remark The above theorem proves a high probability bound on the regret. We can get a bound on the expected regret if we set δ to $1/\sqrt{T}$. Also note that the bound given by the induction grows in the number of non-dominated actions as $O(K^{\log_2 K})$.

4.5 Lower bound for non-trivial games

In the following sections, $\|\cdot\|_1$ and $\|\cdot\|$ denote the L_1- and L_2-norm of a vector in a Euclidean space, respectively.

In this section, we show that non-trivial games have minimax regret at least $\Omega(\sqrt{T})$. We state and prove this result for *all* finite games, in contrast to earlier related lower bounds that apply to specific losses (see Cesa-Bianchi and Lugosi [Cesa-Bianchi and Lugosi, 2006, Theorems 3.7, 6.3, 6.4, 6.11] for full-information, label efficient, and bandit games).

Theorem 4 (Lower bound for non-trivial games)**.** *If $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ is a finite non-trivial ($K \geq 2$) partial-monitoring game then there exists a constant $c > 0$ such that for any $T \geq 1$ the minimax expected regret $\mathrm{R}_T(\mathbf{G}) \geq c\sqrt{T}$.*

We prove the above theorem under two different assumptions. The first proof assumes that the opponent is non-stochastic. In the second proof we lift this assumption and prove the theorem for stochastic opponents. Note that, as opposed to upper bound statements, in the case of lower bounds, bounds stated for the stochastic case are "stronger" in the sense that a lower bound in the stochastic case implies the same bound for the adversarial case.

Proof of Theorem 4 for adversarial opponents. We start with a lemma that ensures the existence of a pair i_1,i_2 of actions and an outcome distribution p with M atoms such that both i_1 and i_2 are optimal under p.

Lemma 4. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be any finite non-trivial game with N actions and $M \geq 2$ outcomes. Then there exists $p \in \Delta_M$ satisfying both of the following properties:*

(a) All coordinates of p are positive.

(b) There exist actions i_1,$i_2 \in \underline{N}$ such that $\ell_{i_1} \neq \ell_{i_2}$ and for all $i \in \underline{N}$,

$$\ell_{i_1}^\top p = \ell_{i_2}^\top p \leq \ell_i^\top p\,.$$

The following lemma is a variant of Khinchine's inequality (see e.g. [Cesa-Bianchi and Lugosi, 2006, Lemma A.9]) for asymmetric random variables. The idea of the proof is the same as there and originally comes from Littlewood [1930].

Lemma 5 (Khinchine's inequality for asymmetric random variables)**.** *Let*

$$X_1, X_2, \ldots, X_T$$

be i.i.d. random variables with mean $\mathbb{E}[X_t] = 0$*, finite variance* $\mathbb{E}[X_t^2] = \mathbb{V}ar[X_t] = \sigma^2$*, and finite fourth moment* $\mathbb{E}[X_t^4] = \mu_4$*. Then,*

$$\mathbb{E}\left|\sum_{t=1}^{T} X_t\right| \geq \frac{\sigma^3}{\sqrt{3\mu_4}}\sqrt{T}\ .$$

When $M = 1$, $\mathbf{G}$ is always trivial, thus we assume that $M \geq 2$. Without loss of generality we may assume that all the actions are all-revealing.

Let $p \in \Delta_M$ be a distribution of the outcomes that satisfies conditions (a) and (b) of Lemma 4. By renaming actions we can assume without loss of generality that $\ell_1 \neq \ell_2$ and actions 1 and 2 are optimal under p, that is,

$$\ell_1^\top p = \ell_2^\top p \leq \ell_i^\top p \tag{4.3}$$

for any $i \in \underline{N}$.

Fix any learning algorithm $\mathcal{A}$. We use randomization, replacing the outcomes with a sequence $J_1, J_2, \ldots, J_T$ of random variables i.i.d. according to p, and independently of the internal randomization of $\mathcal{A}$. Then we can write

$$\begin{aligned}\mathbb{E}[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] &= \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{L}[I_t, J_t] - \min_{i\in\underline{N}}\sum_{t=1}^{T}\mathbf{L}[i, J_t]\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{E}[\mathbf{L}[I_t, J_t] \mid I_t] - \min_{i\in\underline{N}}\sum_{t=1}^{T}\mathbf{L}[i, J_t]\right]\ . \end{aligned} \tag{4.4}$$

Here, in the last two expressions, the expectation is with respect to both the internal randomization of $\mathcal{A}$ and the random choice of $J_1, J_2, \ldots, J_T$. Now, since J_t is independent of I_t, we see that $\mathbb{E}[\mathbf{L}[I_t, J_t] \mid I_t] = \ell_{I_t}^\top p$. By (4.3), we have $\ell_{I_t}^\top p \geq \ell_1^\top p = \ell_2^\top p$. Therefore (upper bounding also the minimum),

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}\left[\mathbf{L}[I_t, J_t] \mid I_t\right] - \min_{i\in\underline{N}}\sum_{t=1}^{T}\mathbf{L}[i, J_t] &= \sum_{t=1}^{T}\ell_{I_t}^\top p - \min_{i\in\underline{N}}\sum_{t=1}^{T}\mathbf{L}[i, J_t] \\ &\geq \sum_{t=1}^{T}\ell_1^\top p - \min_{i=1,2}\sum_{t=1}^{T}\mathbf{L}[i, J_t] \\ &= \max_{i=1,2}\sum_{t=1}^{T}\left(\ell_1^\top p - \mathbf{L}[i, J_t]\right)\ . \end{aligned} \tag{4.5}$$

Using the identity $\max\{a,b\} = \frac{1}{2}(a+b+|a-b|)$, the latest expression is

$$\begin{aligned}
&\frac{1}{2}\Bigg[\sum_{t=1}^{T}\left(\ell_1^\top p - \mathbf{L}[1,J_t]\right) + \sum_{t=1}^{T}\left(\ell_1^\top p - \mathbf{L}[2,J_t]\right) \\
&+ \left|\sum_{t=1}^{T}\left(\ell_1^\top p - \mathbf{L}[1,J_t]\right) - \sum_{t=1}^{T}\left(\ell_1^\top p - \mathbf{L}[2,J_t]\right)\right|\Bigg] \\
&= \frac{1}{2}\sum_{t=1}^{T}\left(\ell_1^\top p - \mathbf{L}[1,J_t] + \ell_2^\top p - \mathbf{L}[2,J_t]\right) + \frac{1}{2}\left|\sum_{t=1}^{T}\left(\mathbf{L}[2,J_t] - \mathbf{L}[1,J_t]\right)\right|,
\end{aligned}$$

where (4.3) was used in the first term. The expectation of the first term vanishes since $\mathbb{E}[\mathbf{L}[i,J_t]] = \ell_i^\top p$. Let $X_t = \mathbf{L}[2,J_t] - \mathbf{L}[1,J_t]$. We see that $X_1, X_2, \ldots, X_T$ are i.i.d. random variables with mean $\mathbb{E}[X_t] = 0$. Therefore,

$$\mathbb{E}\left[\max_{i=1,2}\sum_{t=1}^{T}\left(\ell_1^\top p - \mathbf{L}[i,J_t]\right)\right] = \frac{1}{2}\,\mathbb{E}\left|\sum_{t=1}^{T} X_t\right| \geq c\sqrt{T}, \tag{4.6}$$

where the last inequality follows from Theorem 5 and the constant c depends only on ℓ_1, ℓ_2, and p. For the theorem to yield $c > 0$, it is important to note that the distribution of X_t has finite support and with positive probability $X_t \neq 0$ since $\ell_1 \neq \ell_2$ and all coordinates of p are positive. Hence, both $\mathbb{E}[X_t^2]$ and $\mathbb{E}[X_t^4]$ are finite and positive.

Now, putting together (4.4), (4.5), and (4.6) gives the desired lower bound $\mathbb{E}[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] \geq c\sqrt{T}$. Since c depends only on $\mathbf{L}$, also $\mathrm{R}_T(\mathbf{G}) \geq c\sqrt{T}$.

□

Proof of Theorem 4 for stochastic opponents. The proof is similar to the lower bound proof of Auer et al. [2003].

Recall that $\Delta_M \subset \mathbb{R}^M$ is the $(M-1)$-dimensional probability simplex. For the proof, we start with a geometrical lemma, which ensures the existence of a pair i_1,i_2 of non-dominated actions that are "neighbors" in the sense that for any small enough $\epsilon > 0$, there exists a pair of "ϵ-close" outcome distributions $p + \epsilon w$ and $p - \epsilon w$ such that i_1 is uniquely optimal under the first distribution, and i_2 is uniquely optimal under the second distribution overtaking each non-optimal action by at least $\Omega(\epsilon)$ in both cases.

Lemma 6 (ϵ-close distributions)**.** *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be any finite non-trivial game with N non-duplicate actions and $M \geq 2$ outcomes. Then there exist two non-dominated actions i_1,$i_2 \in \underline{N}$, $p \in \Delta_M$, $w \in \mathbb{R}^M \setminus \{0\}$, and c,$\alpha > 0$ satisfying the following properties:*

(a) $\ell_{i_1} \neq \ell_{i_2}$.

(b) $\ell_{i_1}^\top p = \ell_{i_2}^\top p \leq \ell_i^\top p$ for all $i \in \underline{N}$ and the coordinates of p are positive.

(c) Coordinates of w satisfy $\sum_{j=1}^{M} w(j) = 0$.

For any $\epsilon \in (0, \alpha)$,

(d) $p_1 = p + \epsilon w \in \Delta_M$ *and* $p_2 = p - \epsilon w \in \Delta_M$,

(e) for any $i \in \underline{N}$, $i \neq i_1$, *we have* $(\ell_i - \ell_{i_1})^\top p_1 \geq c\epsilon$,

(f) for any $i \in \underline{N}$, $i \neq i_2$, *we have* $(\ell_i - \ell_{i_2})^\top p_2 \geq c\epsilon$.

We now continue with a technical lemma, which can be used to derive an upper bound on the Kullback-Leibler (KL) divergence (or relative entropy) between the distributions $p-\epsilon w$, $p+\epsilon w$ from the previous lemma. Recall that the KL divergence between two probability distributions p,$q \in \Delta_M$ is defined as

$$D(p \parallel q) = \sum_{j=1}^{M} p_j \ln\left(\frac{p_j}{q_j}\right) .$$

Lemma 7 (KL divergence of ϵ-close distributions)**.** *Let* $p \in \Delta_M$ *be a probability vector and let* $\underline{p} = \min_{j \in \underline{M}: p(j) > 0} p(j)$. *For any vector* $\epsilon \in \mathbb{R}^M$ *such that both* $p - \epsilon$ *and* $p + \epsilon$ *lie in* Δ_M *and* $|\epsilon(j)| \leq p(j)/2$ *for all* $j \in \underline{M}$, *the KL divergence of* $p - \epsilon$ *and* $p + \epsilon$ *satisfies*

$$D(p - \epsilon \parallel p + \epsilon) \leq c\|\epsilon\|^2,$$

where $c = \frac{6\ln(3)-4}{\underline{p}} > 0$.

When $M = 1$, $\mathbf{G}$ is always trivial, thus we assume that $M \geq 2$. Without loss of generality we may assume that all the actions are all-revealing. Then, as in Section 4.2 for M=2, we can also assume that there are no duplicate actions, thus for any two actions i and i', $\ell_i \neq \ell_{i'}$.

Lemma 6 implies that there exist two actions i_1,i_2, $p \in \Delta_M$, $w \in \mathbb{R}^M$, and c_1,$\alpha > 0$ satisfying conditions (a)–(f). To avoid cumbersome indexing, by renaming the actions we can achieve that $i_1 = 1$ and $i_2 = 2$. Let $p_1 = p + \epsilon w$ and $p_2 = p - \epsilon w$ for some $\epsilon \in (0, \alpha)$. We determine the precise value of ϵ later. By Lemma 6 (d), p_1,$p_2 \in \Delta_M$.

Fix any randomized learning algorithm $\mathcal{A}$ and time horizon T. Let

$$J_1, J_2, \ldots, J_T$$

be i.i.d. random variables chosen from p_k with either $k = 1$ or $k = 2$. It is assumed that $(J_1, J_2, \ldots, J_T)$ are independent of the internal randomization of $\mathcal{A}$. For $k \in \{1, 2\}$, let $\mathbb{P}_k(\cdot)$ denote the probability measure induced when $J_t \sim p_k$, while let $\mathbb{E}_k(\cdot)$ be the corresponding expectation operator. Let

$$N_i^{(k)} = N_i^{(k)}(\mathcal{A}, T) = \sum_{t=1}^{T} \mathbb{P}_k\left(I_t = i\right) \in [0, T] \tag{4.7}$$

denote the expected number of times action i is chosen by $\mathcal{A}$ up to time step T when $J_t \sim p_k$.

Parts (e) and (f) of Lemma 6 imply that for any $i \in \underline{N}$ if $\ell_i \neq \ell_k$ then $(\ell_i - \ell_k)^\top p_k \geq c_1\epsilon$. Therefore, we can bound the expected regret as

$$\mathbb{E}_k[R_T^{\mathcal{A}}(G)] = \sum_{i \in \underline{N} \setminus \{k\}} N_i^{(k)} (\ell_i - \ell_k)^\top p_k \geq \sum_{i \in \underline{N} \setminus \{k\}} N_i^{(k)} c_1 \epsilon = c_1 \left(T - N_k^{(k)}\right) \epsilon \,. \tag{4.8}$$

Averaging (4.8) over $k \in \{1, 2\}$ we get

$$\mathbb{E}[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] \geq c_1 \left(2T - N_1^{(1)} - N_2^{(2)}\right) \epsilon/2 \,. \tag{4.9}$$

We now focus on lower bounding $2T - N_1^{(1)} - N_2^{(2)}$. We start by showing that $N_2^{(2)}$ is close to $N_2^{(1)}$. The following lemma, which is the key lemma of both lower bound proofs, carries this out formally and states that the expected number of times an action is played by $\mathcal{A}$ does not change too much when we change the model, if the outcome distributions p_1 and p_2 are "close" in KL-divergence:

Lemma 8. *For any partial-monitoring game with N actions and M outcomes, algorithm $\mathcal{A}$, pair of outcome distributions p_1,$p_2 \in \Delta_M$ and action i, we have*

$$N_i^{(2)} - N_i^{(1)} \leq T\sqrt{D(p_2 \parallel p_1) N_{\mathrm{rev}}^{(2)}/2}$$

and

$$N_i^{(1)} - N_i^{(2)} \leq T\sqrt{D(p_1 \parallel p_2) N_{\mathrm{rev}}^{(1)}/2},$$

where $N_{\mathrm{rev}}^{(k)} = \sum_{t=1}^{T} \mathbb{P}_k\left(I_t \in \mathcal{R}\right) = \sum_{i' \in \mathcal{R}} N_{i'}^{(k)}$ under model p_k, $k = 1{,}2$ with $\mathcal{R}$ being the set of revealing actions.[5]

We use Lemma 8 for $i = 2$ and that $N_{\mathrm{rev}}^{(2)} \leq T$ to bound the difference $N_2^{(2)} - N_2^{(1)}$ as

$$N_2^{(2)} - N_2^{(1)} \leq T\sqrt{D(p_2 \parallel p_1)T/2} = T^{3/2}\sqrt{D(p_2 \parallel p_1)/2} \,. \tag{4.10}$$

We upper bound $D(p_2 \parallel p_1)$ using Lemma 7 with $\epsilon = \epsilon w$. The lemma implies that $D(p_2 \parallel p_1) \leq c_2\epsilon^2$ for $\epsilon < \epsilon_0$ with some ϵ_0, $c_2 > 0$ which depend only on w and p. Putting this together with (4.10) we get

$$N_2^{(2)} < N_2^{(1)} + c_3 \epsilon T^{3/2}$$

where $c_3 = \sqrt{c_2/2}$. Together with $N_1^{(1)} + N_2^{(1)} \leq T$ we get

$$2T - N_1^{(1)} - N_2^{(2)} > 2T - N_1^{(1)} - N_2^{(1)} - c_3\epsilon T^{3/2} \geq T - c_3\epsilon T^{3/2} \,.$$

[5] It seems from the proof that $N_{\mathrm{rev}}^{(k)}$ could be slightly sharpened to $N_{\mathrm{rev}}^{(k,T-1)} = \sum_{t=1}^{T-1} \mathbb{P}_k\left(I_t \in \mathcal{R}\right)$.

Substituting into (4.9) and choosing $\epsilon = 1/(2c_3T^{1/2})$ gives the desired lower bound

$$\mathbb{E}[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] > \frac{c_1}{8c_3}\sqrt{T}$$

provided that our choice of ϵ ensures that $\epsilon < \min(\alpha, \epsilon_0) =: \epsilon_1$ that depends only on $\mathbf{L}$. This condition is satisfied for all $T > T_0 = 1/(2c_3\epsilon_1)^2$. Since c_1, c_3, and ϵ_1 depend only on $\mathbf{L}$, for such T, $\mathrm{R}_T(\mathbf{G}) \geq \frac{c_1}{8c_3}\sqrt{T}$.

The non-triviality of the game implies that Lemma 1 d) does not hold, so neither does b), that is, $\mathrm{R}_T(\mathbf{G}) > 0$ for $T \geq 1$. Thus choosing

$$c = \min\left(\min_{1\leq T\leq T_0} \frac{\mathrm{R}_T(\mathbf{G})}{\sqrt{T}}, \frac{c_1}{8c_3}\right),$$

$c > 0$ and for any T, $\mathrm{R}_T(\mathbf{G}) \geq c\sqrt{T}$. □

4.6 Lower bound for hard games

In this section, we present an $\Omega(T^{2/3})$ lower bound for the expected regret of any two-outcome game in the case when the separation condition does not hold.

Theorem 5 (Lower bound for hard games)**.** *If $M = 2$ and $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ satisfies the non-degeneracy condition and the separation condition does **not** hold then there exists a constant $C > 0$ such that for any $T \geq 1$ the minimax expected regret $\mathrm{R}_T(\mathbf{G}) \geq CT^{2/3}$.*

Proof. We follow the lower bound proof for the label efficient prediction from Cesa-Bianchi and Lugosi [2006] with a few changes. The most important change, as we will see, is the choice of the models we randomize over.

As the first step, the following lemma shows that non-revealing degenerate actions do not influence the minimax regret of a game.

Lemma 9. *Let $\mathbf{G}$ be a non-degenerate game with two outcomes. Let $\mathbf{G}'$ be the game we get by removing the degenerate non-revealing actions from $\mathbf{G}$. Then $\mathrm{R}_T(\mathbf{G}) = \mathrm{R}_T(\mathbf{G}')$.*

By the non-degeneracy condition and Lemma 9, we can assume without loss of generality that $\mathbf{G}$ does not have degenerate actions. We can also assume without loss of generality that actions 1 and 2 are the two consecutive non-dominated non-revealing actions. It follows by scaling and a reduction similar to the one we used in Section 4.4.1 that we can further assume $(\mathbf{L}[1,1], \mathbf{L}[1,2]) = (0, \alpha)$, $(\mathbf{L}[2,1], \mathbf{L}[2,2]) = (1-\alpha, 0)$ with some $\alpha \in (0,1)$. Using the non-degeneracy condition and that actions 1 and 2 are consecutive non-dominated actions, we get that for all $i \geq 3$, there exists some $\lambda_i \in \mathbb{R}$ depending only on $\mathbf{L}$ such that

$$\begin{aligned} \mathbf{L}[i,1] &> \lambda_i\mathbf{L}[1,1] + (1-\lambda_i)\mathbf{L}[2,1] = (1-\lambda_i)(1-\alpha)\ , \\ \mathbf{L}[i,2] &> \lambda_i\mathbf{L}[1,2] + (1-\lambda_i)\mathbf{L}[2,2] = \lambda_i\alpha\ . \end{aligned} \tag{4.11}$$

Let $\lambda_{\min} = \min_{i\geq 3} \lambda_i$, $\lambda_{\max} = \max_{i\geq 3} \lambda_i$, and $\lambda^* = \lambda_{\max} - \lambda_{\min}$.

We define two models for generating outcomes from $\{1, 2\}$. In model 1, the outcome distribution is $p_1(1) = \alpha + \epsilon$, $p_1(2) = 1 - p_1(1)$, whereas in model 2, $p_2(1) = \alpha - \epsilon$, $p_2(2) = 1 - p_2(1)$ with $0 < \epsilon \leq \min(\alpha, 1 - \alpha)/2$ to be chosen later.

Let $J_1, J_2, \ldots, J_T$ be i.i.d. random variables chosen from p_k with either $k = 1$ or $k = 2$. It is assumed that $(J_1, J_2, \ldots, J_T)$ are independent of the internal randomization of $\mathcal{A}$. For $k \in \{1, 2\}$, let $\mathbb{P}_k(\cdot)$ denote the probability measure induced when $J_t \sim p_k$, while let $\mathbb{E}_k(\cdot)$ be the corresponding expectation operator.

Let $N_i^{(k)}$ be the expected number of times action i is chosen by $\mathcal{A}$ under p_k up to time step T, as in (4.7).

Finally, let $N_{\geq 3}^{(k)} = \sum_{i\geq 3} N_i^{(k)}$. Note that, if $\epsilon < \epsilon_0$ with some ϵ_0 depending only on $\mathbf{L}$ then only actions 1 and 2 can be optimal for these models. Namely, action k is optimal under p_k, hence $\mathbb{E}_k[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})]$ can be expressed in terms of $N_i^{(k)}$:

$$\mathbb{E}_k[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] = \sum_{i\in\underline{N}\setminus\{k\}} N_i^{(k)}(\ell_i - \ell_k)^\top p_k = \sum_{i=3}^{N} N_i^{(k)}(\ell_i - \ell_k)^\top p_k + N_{3-k}^{(k)}(\ell_{3-k} - \ell_k)^\top p_k \tag{4.12}$$

for $k = 1,2$. Now, by (4.11), there exists $\tau > 0$ depending only on $\mathbf{L}$ such that for all $i \geq 3$, $\mathbf{L}[i, 1] \geq (1 - \lambda_i)(1 - \alpha) + \tau$ and $\mathbf{L}[i, 2] \geq \alpha\lambda_i + \tau$. These bounds and simple algebra give that

$$\begin{aligned}
(\ell_i - \ell_1)^\top p_1 &= (\mathbf{L}[i, 1] - \mathbf{L}[1, 1])(\alpha + \epsilon) + (\mathbf{L}[i, 2] - \mathbf{L}[1, 2])(1 - \alpha - \epsilon) \\
&\geq ((1 - \lambda_i)(1 - \alpha) + \tau)(\alpha + \epsilon) + (\alpha\lambda_i + \tau - \alpha)(1 - \alpha - \epsilon) \\
&= (1 - \lambda_i)\epsilon + \tau \\
&\geq (1 - \lambda_{\max})\epsilon + \tau =: f_1
\end{aligned}$$

and

$$(\ell_2 - \ell_1)^\top p_1 = (1 - \alpha)(\alpha + \epsilon) - \alpha(1 - \alpha - \epsilon) = \epsilon\,.$$

Analogously, we get

$$(\ell_i - \ell_2)^\top p_2 \geq \lambda_{\min}\epsilon + \tau =: f_2 \qquad \text{and} \qquad (\ell_1 - \ell_2)^\top p_2 = \epsilon\,.$$

Note that if $\epsilon < \tau/\max(|1 - \lambda_{\max}|, |\lambda_{\min}|)$ then both f_1 and f_2 are positive. Substituting these into (4.12) gives

$$\mathbb{E}_k[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] \geq f_k N_{\geq 3}^{(k)} + \epsilon N_{3-k}^{(k)}\,. \tag{4.13}$$

The following lemma is an application of Lemmas 8 and 7:

Lemma 10. *There exists a constant $c > 0$ (depending on α only) such that for any $\epsilon > 0$,*

$$N_2^{(1)} \geq N_2^{(2)} - cT\epsilon\sqrt{N_{\geq 3}^{(2)}} \qquad \textit{and} \qquad N_1^{(2)} \geq N_1^{(1)} - cT\epsilon\sqrt{N_{\geq 3}^{(1)}}\,.$$

Let $l = \arg\min_{k\in\{1,2\}} N^{(k)}_{\geq 3}$. Now, for $k \neq l$ we can lower bound the regret using Lemma 10 for (4.13):

$$\mathbb{E}_k[\mathrm{R}^{\mathcal{A}}_T(\mathbf{G})] \geq f_k N^{(k)}_{\geq 3} + \epsilon\left(N^{(l)}_{3-k} - cT\epsilon\sqrt{N^{(l)}_{\geq 3}}\right) \geq f_k N^{(l)}_{\geq 3} + \epsilon\left(N^{(l)}_{3-k} - cT\epsilon\sqrt{N^{(l)}_{\geq 3}}\right) \tag{4.14}$$

as $f_k > 0$. For $k = l$ we do this subtracting $cT\epsilon^2\sqrt{N^{(l)}_{\geq 3}} \geq 0$ from the right-hand side of (4.13) leading to the same lower bound, hence (4.14) holds for k = 1,2. Finally, averaging (4.14) over $k \in \{1, 2\}$ we have the bound

$$\begin{aligned}
\frac{f_1 + f_2}{2} N^{(l)}_{\geq 3} + \epsilon\left(\frac{N^{(l)}_2 + N^{(l)}_1}{2} - cT\epsilon\sqrt{N^{(l)}_{\geq 3}}\right) &= \left(\frac{(1 - \lambda_{\max} + \lambda_{\min})\epsilon}{2} + \tau\right) N^{(l)}_{\geq 3} \\
&\quad + \epsilon\left(\frac{T - N^{(l)}_{\geq 3}}{2}\right) - cT\epsilon^2\sqrt{N^{(l)}_{\geq 3}} \\
&= \left(\tau - \frac{\lambda^*\epsilon}{2}\right) N^{(l)}_{\geq 3} + \frac{\epsilon T}{2} - cT\epsilon^2\sqrt{N^{(l)}_{\geq 3}}\,.
\end{aligned}$$

Choosing $\epsilon = c_2 T^{-1/3}$ ($\leq c_2$) with $c_2 > 0$ gives

$$\begin{aligned}
\mathbb{E}[\mathrm{R}^{\mathcal{A}}_T(\mathbf{G})] &\geq \left(\tau - \frac{\lambda^* c_2 T^{-1/3}}{2}\right) N^{(l)}_{\geq 3} + \frac{c_2 T^{2/3}}{2} - cc_2^2 T^{1/3}\sqrt{N^{(l)}_{\geq 3}} \\
&\geq \left(\tau - \frac{\lambda^* c_2}{2}\right) N^{(l)}_{\geq 3} + \frac{c_2 T^{2/3}}{2} - cc_2^2 T^{1/3}\sqrt{N^{(l)}_{\geq 3}} \\
&= \left(\left(\tau - \frac{\lambda^* c_2}{2}\right) x^2 + \frac{c_2}{2} - cc_2^2 x\right) T^{2/3} = q(x) T^{2/3},
\end{aligned}$$

where $x = T^{-1/3}\sqrt{N^{(l)}_{\geq 3}}$ and $q(x)$ can be written and lower bounded as

$$q(x) = \left(\tau - \frac{\lambda^* c_2}{2}\right)\left(x - \frac{cc_2^2}{2\tau - \lambda^* c_2}\right)^2 + \frac{c_2}{2} - \frac{c^2 c_2^4}{4\tau - 2\lambda^* c_2} \geq \frac{c_2}{2}\left(1 - \frac{c^2 c_2}{2\tau - \lambda^* c_2}\right)$$

independently of x whenever $\lambda^* c_2 < 2\tau$ and $c_2 \leq 1$. Now, it is easy to see that if $c_2 = \min(\tau/(c^2 + \lambda^*), 1)$ then these hold, moreover, $q(x) \geq c_2/4 > 0$ giving the desired lower bound

$$\mathbb{E}[\mathrm{R}^{\mathcal{A}}_T(\mathbf{G})] \geq \frac{c_2}{4} T^{2/3}$$

provided that our choice of ϵ ensures that $\epsilon < \min(\alpha/2, (1-\alpha)/2, \epsilon_0, \tau/|1 - \lambda_{\max}|, \tau/|\lambda_{\min}|) =: \epsilon_1$ that depends only on $\mathbf{L}$. This condition is satisfied for all $T > T_0 = (c_2/\epsilon_1)^3$. Since c_2 and ϵ_1 depend only on $\mathbf{L}$, for such T, $\mathrm{R}_T(\mathbf{G}) \geq \frac{c_2}{4} T^{2/3}$.

If the separation condition does not hold then the game is clearly non-trivial which, using Lemma 1 b) and d) as in the proof of Theorem 4, implies that $\mathrm{R}_T(\mathbf{G}) > 0$ for $T \geq 1$. Thus choosing

$$C = \min\left(\min_{1\leq T\leq T_0} \frac{\mathrm{R}_T(\mathbf{G})}{T^{2/3}}, \frac{c_2}{4}\right),$$

$C > 0$ and for any T, $\mathrm{R}_T(\mathbf{G}) \geq CT^{2/3}$. □

4.7 Discussion

In this chapter we classified non-degenerate partial-monitoring games with two outcomes based on their minimax regret. An immediate question is how the classification extends to degenerate games. From the results in this chapter, we can not even tell if all degenerate games fall into one of the four categories or whether there are some games with minimax regret of $\widetilde{\Theta}(T^\alpha)$ for some $\alpha \in (1/2, 2/3)$.

Besides the issue of degenerate games, the most important question is whether the results generalize to games with more outcomes. A simple observation is that, given a finite partial-monitoring game, if we restrict the opponent's choices to any two outcomes, the resulting game's hardness serves as a lower bound on the minimax regret of the original game. This gives us a sufficient condition that a game has $\Omega(T^{2/3})$ minimax regret.

As it turns out, the separation condition—the condition that separates easy from hard games with two outcomes—can be generalized to general finite games. We explain how the generalization is carried out in Chapter 6, where we give the classification (including degenerate games) for all finite games against stochastic environments. The question against adversarial opponents, building upon the work presented in Chapter 6, was answered by Foster and Rakhlin [2011] (for a short summary of their work, see Section 6.2.5).

Chapter 5

Two-action games[1]

In the previous chapter, we dealt with games with two outcomes. The analysis of such games led us to the observation that two-action games play an important rule in the classification. Indeed, the algorithm AppleTree splits the game into two-action games, and then plays those games depending on their feedback structure. For that reason, we started to think about the "dual" case: games with two actions and any finite number of outcomes. We know from the previous chapter that to have a $\Omega(T^{2/3})$ lower bound on the minimax regret of a game with two outcomes, one needs at least three actions: two consecutive non-revealing actions are needed so that the game is not easy, and a revealing action to make the game non-hopeless. This naturally raises questions: If a game has only two actions, what can its minimax regret be? Do we have the same four classes as in the two-outcome case? Or is the "hard" class missing? Are there more classes due to more outcomes? In this chapter, we answer these questions.

5.1 Results

The intuition that having $\Theta(T^{2/3})$ minimax regret requires at least three actions turns out to be a good lead: in this chapter we show that for two-action games, only three classes exist. These three classes are: trivial, easy, and hopeless. In fact, we show something even stronger: we prove that if a game is not trivial nor hopeless, then it can be transformed to a bandit game. Then any bandit algorithm can be used to have $O(\sqrt{T})$ regret.

To prove the above statement, we first need to precisely define what we mean by "transforming" a game to another.

Definition 4. *Take two games,* $\mathbf{G} = (\mathbf{L}, \mathbf{H})$, $\mathbf{G}' = (\mathbf{L}', \mathbf{H}')$, *where* $\mathbf{L}$, $\mathbf{L}'$, $\mathbf{H}$, *and* $\mathbf{H}'$ *are* $N \times M$ *matrices. We say that* $\mathbf{G}'$ *is* simulation-and-regret-not-harder *than* $\mathbf{G}$ *(or, in short,* $\mathbf{G}'$ *is* easier *than* $\mathbf{G}$*, or* $\mathbf{G}' \leq \mathbf{G}$*) when the following holds: Fix any algorithm* $\mathcal{A}$*. Then, one can find an algorithm* $\mathcal{A}'$ *such that the behavior of* $\mathcal{A}$ *on* $\mathbf{G}$ *can be replicated by using* $\mathcal{A}'$ *on* $\mathbf{G}'$ *in the*

[1]Based on joint work with András Antos and Csaba Szepesvári.

sense that for the same outcome sequence, the two algorithms will choose the same action sequences and the regret in the second case is at most the regret in the first case, that is, $\mathrm{R}_T^{\mathcal{A}'}(\mathbf{G}') \le \mathrm{R}_T^{\mathcal{A}}(\mathbf{G})$.

We say that $\mathbf{G}$ *and* $\mathbf{G}'$ *are* simulation-and-regret-equivalent *(or* equivalent*,* $\mathbf{G}' \simeq \mathbf{G}$*) when both* $\mathbf{G}' \le \mathbf{G}$ *and* $\mathbf{G} \le \mathbf{G}'$.

Clearly, $\le$ is a preorder and $\simeq$ is an equivalence relation on the set of $N \times M$ games, moreover, if $\mathbf{G}' \le \mathbf{G}$ then $\mathrm{R}_T(\mathbf{G}') \le \mathrm{R}_T(\mathbf{G})$, and if $\mathbf{G} \simeq \mathbf{G}'$ then their minimax regret is the same.

We need a few simple lemmata on these relations of games:

Lemma 11. *The regret of a sequence of actions in a game* $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ *does not change if the loss matrix is changed by subtracting the same real number from each coordinate of one of its columns (see e.g., Piccolboni and Schindelhauer [2001]). Therefore, letting* $\mathbf{1} = (1, \dots, 1)^\top \in \mathbb{R}^N$*,* $\mathbf{v} \in \mathbb{R}^M$*, and* $\mathbf{G}' = (\mathbf{L} - \mathbf{1}\mathbf{v}^\top, \mathbf{H})$*, we have that* $\mathbf{G} \simeq \mathbf{G}'$.

Lemma 12. *If* $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ *and* $\mathbf{G}' = (\mathbf{L}, \mathbf{H}')$ *differ only in their feedback matrices and* $\mathbf{H}'$ *can be obtained by* $h'_{ij} = f_i(h_{ij})$ *with the help of some mappings* f_i *(*$i \in \underline{N}$*) then* $\mathbf{G} \le \mathbf{G}'$*. If each* f_i *is injective then* $\mathbf{G} \simeq \mathbf{G}'$.

In what follows, a transformation of some game into another game that takes either the first or the second form just defined shall be called an *admissible* transformation.

The following proposition shows that if a 2-action partial-monitoring game is non-trivial and non-hopeless then there is no loss in generality by assuming that $\mathbf{L} = \mathbf{K}\mathbf{H}$ for some $\mathbf{K} \in \mathbb{R}^{2\times 2}$. This statement for arbitrary N and most of the ideas for its proof could be extracted from the paper of Piccolboni and Schindelhauer [2001, see Section 4, Theorem 3]. An exact detailed proof for $N = 2$ is included here for the sake of completeness.

Proposition 1. *Let* $\mathbf{G}_0 = (\mathbf{L}_0, \mathbf{H}_0)$ *be a non-trivial non-hopeless 2-action partial-monitoring game. Then, there exist matrices* $\mathbf{L}, \mathbf{H} \in \mathbb{R}^{2\times M}$ *such that* $\mathbf{G}_0 \le \mathbf{G} = (\mathbf{L}, \mathbf{H})$ *and* $\mathbf{L} = \mathbf{K}\mathbf{H}$ *for some* $\mathbf{K} \in \mathbb{R}^{2\times 2}$*. Namely,* $\mathbf{K}$ *can be*

$$\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}. \tag{5.1}$$

Proof of Proposition 1. First, we transform $\mathbf{L}_0$ to $\mathbf{L}$ using Lemma 11 with $\mathbf{v}^\top$ being the first row of $\mathbf{L}_0$. Thus, the first row of $\mathbf{L}$ becomes identically zero, and we get a non-trivial non-hopeless game $\mathbf{G}_1 = (\mathbf{L}, \mathbf{H}_0) \simeq \mathbf{G}_0$. Let ℓ denote the transpose of the second row of $\mathbf{L}$. In what follows we construct the matrix $\mathbf{H}$ using an admissible transformation of $\mathbf{H}_0$ defined in Lemma 12.

To go on with constructing $\mathbf{H}$, we need the following concept:

Definition 5. *Let* $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ *a finite partial-monitoring game. Let* s_i *be the number of distinct symbols in the* i^{th} *row of* $\mathbf{H}$ *and let* $\sigma_1, \dots, \sigma_{s_i} \in \Sigma$ *be an enumeration of those symbols. Then the* signal matrix $S_i \in \{0,1\}^{s_i \times M}$ *of action* i *is defined as* $S_i[k, l] = \mathbb{I}_{\{\mathbf{H}[i,l] = \sigma_k\}}$.

$$\mathbf{H}_0 = \begin{pmatrix} 1 & 2 & 3 & 1 \\ 1 & 2 & 2 & 2 \end{pmatrix} \longrightarrow \mathbf{A} = \left(\begin{array}{cccc} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{array}\right)$$

Figure 5.1: An example for the construction of matrix $\mathbf{A}$ used in the proof of Proposition 1. The first three rows of $\mathbf{A}$ are constructed from the first row of $\mathbf{H}_0$ which has three distinct elements, the remaining two rows are constructed from the second row of $\mathbf{H}_0$. For more details, see the text.

Note that this concept will be extensively used in the next chapter, and thus Definition 5 will be repeated there.

Now, we construct matrix $\mathbf{A}$ in the following way. We take game $\mathbf{G}_0$ and construct the signal matrices S_1 and S_2 according to Definition 5. Then, we define $\mathbf{A}$ by "stacking" the matrices S_1 and S_2 on top of one another:

$$\mathbf{A} = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}.$$

See Figure 5.1 for an example.

The following lemma is the key to prove Proposition 1. We let $\operatorname{Im} M$ denote the *image space* of the matrix M, that is, $\operatorname{Im} M = \{v \ : \ \exists w \text{ s.t. } v = Mw\}$.

Lemma 13. *If $\ell \notin \operatorname{Im} \mathbf{A}^\top$ then $\mathbf{G}_1$ is trivial or hopeless.*

Using the assumption that $\mathbf{G}_1$ is non-trivial and non-hopeless, we have from Lemma 13 that $\ell \in \operatorname{Im} \mathbf{A}^\top$ must hold. That is, ℓ can be written as a linear combination of the rows of $\mathbf{A}$:

$$\ell = \sum_{i=1}^{m} \lambda_i \mathbf{a}_i,$$

where $m = m_1 + m_2$ and the vectors $\mathbf{a}_i^\top$ are the rows of $\mathbf{A}$. Let

$$\mathbf{h}_1 = \sum_{i=1}^{m_1} \lambda_i \mathbf{a}_i \qquad \text{and} \qquad \mathbf{h}_2 = \sum_{i=m_1+1}^{m} \lambda_i \mathbf{a}_i.$$

Finally, let

$$\mathbf{H} = \begin{pmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \end{pmatrix}$$

and $\mathbf{G} = (\mathbf{L}, \mathbf{H})$. Now if the k^{th} and k'^{th} entries of the first row of $\mathbf{H}_0$ are identical then $[\mathbf{a}_i]_k = [\mathbf{a}_i]_{k'}$ for $1 \le i \le m_1$, hence also $[\mathbf{h}_1]_k = [\mathbf{h}_1]_{k'}$. The same holds for the second row of $\mathbf{H}_0$ and $\mathbf{h}_2$. Thus, $\mathbf{H}$ can be obtained by appropriate mappings from $\mathbf{H}_0$, and Lemma 12 implies $\mathbf{G}_1 \le \mathbf{G}$.

On the other hand, setting $\mathbf{K}$ as in (5.1), $\ell = \mathbf{h}_1 + \mathbf{h}_2$ implies that $\mathbf{L} = \mathbf{KH}$. □

The following Proposition is more than what we need, but it is interesting in itself:

Theorem 6. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be a 2-action partial-monitoring game such that $\mathbf{L} = \mathbf{K}\mathbf{H}$ for some $\mathbf{K} \in \mathbb{R}^{2\times 2}$. Then, there exist a $2 \times M$ bandit game $\mathbf{G}'$ such that $\mathbf{G} \le \mathbf{G}'$. If $\mathbf{K}$ is given by (5.1) then $\mathbf{G} \simeq \mathbf{G}'$.*

Proof. We will construct a bandit game $\mathbf{G}' = (\mathbf{L}', \mathbf{H}') \ge \mathbf{G}$ that satisfies $\mathbf{L}' = \mathbf{H}'$. Let $\mathbf{K} = [k_{ij}]_{2\times 2}$ and

$$\mathbf{D} = \mathrm{diag}(k_{11} - k_{21}, k_{22} - k_{12})$$

be a 2×2 diagonal matrix, and define the feedback matrix of $\mathbf{G}'$ by $\mathbf{H}' = \mathbf{D}\mathbf{H}$. Then, both rows of $\mathbf{H}'$ are scalar multiples of the corresponding rows of $\mathbf{H}$. Hence, by these mappings and Lemma 12, $\mathbf{G} \le (\mathbf{L}, \mathbf{H}')$. If $\mathbf{K}$ is given by (5.1) then $\mathbf{D} = \mathrm{diag}(-1, 1)$, thus both mappings are injective and $\mathbf{G} \simeq (\mathbf{L}, \mathbf{H}')$. On the other hand, $\mathbf{K} - \mathbf{D} = \mathbf{1}\mathbf{k}^\top$ where $\mathbf{k}^\top = (k_{21}, k_{12})$. Consider the loss matrix

$$\mathbf{L}' \stackrel{\triangle}{=} \mathbf{L} - \mathbf{1}(\mathbf{k}^\top \mathbf{H}).$$

By Lemma 11, $\mathbf{G}' = (\mathbf{L}', \mathbf{H}') \simeq (\mathbf{L}, \mathbf{H}')$. Moreover,

$$\mathbf{L}' = \mathbf{L} - (\mathbf{1}\mathbf{k}^\top)\mathbf{H} = \mathbf{L} - (\mathbf{K} - \mathbf{D})\mathbf{H} = \mathbf{D}\mathbf{H} = \mathbf{H}'. \qquad \square$$

Now, we are ready to prove our main result.

Theorem 7. *Each non-trivial non-hopeless 2-action partial-monitoring game is easier than an appropriate $2 \times M$ bandit game. Consequently, its minimax regret is $\Theta(\sqrt{T})$, where T is the number of time steps.*

Proof. According to Proposition 1 and Theorem 6, if $\mathbf{G}_0$ is non-trivial and non-hopeless then we can construct first $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ such that $\mathbf{L} = \mathbf{K}\mathbf{H}$ and $\mathbf{G}_0 \le \mathbf{G}$, then a $2 \times M$ bandit game $\mathbf{G}'$ such that $\mathbf{G} \le \mathbf{G}'$. Thus $\mathbf{G}_0 \le \mathbf{G}'$, which in turn implies $\mathrm{R}_T(\mathbf{G}_0) \le \mathrm{R}_T(\mathbf{G}') = O(\sqrt{T})$ by Auer et al. [2003]. On the other hand, $\mathrm{R}_T(\mathbf{G}_0) = \Omega(\sqrt{T})$ comes from Theorem 4, finishing the proof. $\square$

Remark 1. *It is worthwhile to consider why the above proof works only for $N = 2$. We used the property that from any 2×2 matrix $\mathbf{K}$ we can subtract a diagonal matrix resulting in a matrix with identical rows. For $N \ge 3$, this obviously does not hold (there is not enough "degrees of freedom"). Indeed, for $N \ge 3$, we know from Chapter 4 that there exist games with regret rates between $\Theta(\sqrt{T})$ and $\Theta(T)$.*

The immediate implication of Theorem 7 is the following classification theorem:

Theorem 8. *Every two-action finite partial-monitoring game $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ falls into one of the following three categories*

1. trivial *with minimax regret* 0*;*
2. hopeless *with minimax regret* $\Theta(T)$*; or*
3. bandit-like *with minimax regret* $\Theta(\sqrt{T})$.

5.2 Discussion

In this chapter we showed that if a game has only two actions, then there are only three categories regarding the growth rate of the minimax regret: the class of "hard" games is missing. In some sense, this result is stronger than just classifying all games: we also showed that the games that fall into the "easy" category, are always equivalent to a bandit game. Hence, we did not have to design a new algorithm to prove the $O(\sqrt{T})$ upper bound: one can use any bandit algorithm of their taste, provided that it achieves the desired regret bound.

Chapter 6

Classification of finite stochastic partial-monitoring games[1]

In the two previous chapters, we addressed the problem of classifying partial-monitoring games with constraints on the number of outcomes or actions. In this chapter we turn our attention to games with any finite number of actions and outcomes. We show that against *stochastic* opponents, the same four categories are present as for two-outcome games, including even the degenerate games.

6.1 Preliminaries

Recall that an instance of partial monitoring with N actions and M outcomes is defined by the pair of matrices $\mathbf{L} \in \mathbb{R}^{N\times M}$ and $\mathbf{H} \in \Sigma^{N\times M}$, where Σ is an arbitrary set of symbols. In each round t, the opponent chooses an outcome $J_t \in \underline{M}$ and simultaneously the learner chooses an action $I_t \in \underline{N}$. Then, the feedback $\mathbf{H}[I_t, J_t]$ is revealed and the learner suffers the loss $\mathbf{L}[I_t, J_t]$. It is important to note that the loss is not revealed to the learner.

In this chapter we deal with stochastic opponents only. In this case, the outcome sequence $J_1, J_2, \ldots$ is an i.i.d. sequence of random variables. The common distribution of these random variables, $p \in \Delta_M$, shall be called an *opponent strategy*, where Δ_M, also called the probability simplex, is the set of all distributions over the M outcomes. Given an opponent strategy p, the expected loss of action i equals to $\ell_i^\top p$, where ℓ_i is the column vector obtained from the i^{th} row of $\mathbf{L}$.

The following definitions are essential for understanding how the structure of $\mathbf{L}$ and $\mathbf{H}$ determines the "hardness" of a game.

Action i is called *optimal* under strategy p if its expected loss is not greater than that of any other action $i' \in \underline{N}$. That is, $\ell_i^\top p \leq \ell_{i'}^\top p$. Determining which actions are optimal under the various opponent strategies yields the *cell decomposition*[2] of the probability simplex Δ_M:

[1] A version of the work in this chapter appeared in Bartók, Pál, and Szepesvári [2011]

[2] The concept of cell decomposition also appears in Piccolboni and Schindelhauer [2001].

Definition 6 (Cell decomposition)**.** *For every action $i \in \underline{N}$, let $C_i = \{p \in \Delta_M : \text{action } i \text{ is optimal under } p\}$. The sets $C_1, \ldots, C_N$ constitute the* cell decomposition *of Δ_M.*

Now we can define the following important properties of actions:

Definition 7 (Properties of actions)**.**

- *Action i is called* dominated *if $C_i = \emptyset$. If an action is not dominated then it is called* non-dominated*.*
- *Action i is called* degenerate *if it is non-dominated and there exists an action i' such that $C_i \subsetneq C_{i'}$.*
- *If an action is neither dominated nor degenerate then it is called* Pareto-optimal*. The set of Pareto-optimal actions is denoted by $\mathcal{P}$.*

From the definition of cells, we can see that the cell of an action is either empty or it is a closed convex polytope. Furthermore, Pareto-optimal actions have $(M-1)$-dimensional cells. The following definition also uses the dimensionality of polytopes:

Definition 8 (Neighbors)**.** *Two Pareto-optimal actions i and j are* neighbors *if $C_i \cap C_j$ is an $(M-2)$-dimensional polytope. We denote by $\mathcal{N}$ the set of unordered pairs over $\underline{N}$ that contains neighboring action-pairs. The* neighborhood action set *of two neighboring actions i, j is defined as $N^+_{i,j} = \{k \in \underline{N} : C_i \cap C_j \subseteq C_k\}$.*

Note that the neighborhood action set $N^+_{i,j}$ naturally contains i and j. If $N^+_{i,j}$ contains some other action k then either $C_k = C_i$, $C_k = C_j$, or $C_k = C_i \cap C_j$.

In general, the elements of the feedback matrix $\mathbf{H}$ can be arbitrary symbols. Therefore, the nature of the symbols themselves does not matter in terms of the structure of the game. What determines the feedback structure of a game is the occurrence of identical symbols in each row of $\mathbf{H}$. To "standardize" the feedback structure, we can use the signal matrices introduced in Chapter 5. The definition provided there is repeated to ease the job of the reader:

Definition 5. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ a finite partial-monitoring game. Let s_i be the number of distinct symbols in the i^{th} row of $\mathbf{H}$ and let $\sigma_1, \ldots, \sigma_{s_i} \in \Sigma$ be an enumeration of those symbols. Then the* signal matrix *$S_i \in \{0,1\}^{s_i \times M}$ of action i is defined as $S_i[k,l] = \mathbb{I}_{\{\mathbf{H}[i,l]=\sigma_k\}}$.*

The idea of this definition is that if $p \in \Delta_M$ is the opponent's strategy then $S_i p$ gives the distribution over the symbols underlying action i. In fact, it is also true that observing $\mathbf{H}[I_t, J_t]$ is equivalent to observing the vector $S_{I_t} e_{J_t}$, where e_k is the k^{th} unit vector in the standard basis of $\mathbb{R}^M$. From now on we assume without loss of generality that the learner's observation at time step t

is the random vector $Y_t = S_{I_t} e_{J_t}$. Note that the dimensionality of this vector depends on the action chosen by the learner, namely $Y_t \in \mathbb{R}^{s_{I_t}}$.

Let the symbol $\oplus$ denote the direct sum of subsets of a vector space, that is, $A \oplus B = \{u + v \ : \ u \in A, v \in B\}$. The following two definitions play a key role in classifying partial-monitoring games based on their difficulty.

Definition 9 (Global observability [Piccolboni and Schindelhauer, 2001])**.** *A partial-monitoring game* $(\mathbf{L}, \mathbf{H})$ *admits the* global observability *condition, if for all pairs* i, j *of actions,* $\ell_i - \ell_j \in \oplus_{k \in \underline{N}} \operatorname{Im} S_k^\top$.

Definition 10 (Local observability)**.** *A pair of neighboring actions* i, j *is said to be locally observable if* $\ell_i - \ell_j \in \oplus_{k \in N_{i,j}^+} \operatorname{Im} S_k^\top$. *We denote by* $\mathcal{L} \subset \mathcal{N}$ *the set of locally observable pairs of actions (the pairs are unordered). A game satisfies the* local observability *condition if every pair of neighboring actions is locally observable, i.e., , if* $\mathcal{L} = \mathcal{N}$.

When discussing lower bounds we will need the definition of algorithms. For us, an algorithm $\mathcal{A}$ is a mapping $\mathcal{A} : \Sigma^* \to \{1, 2, \ldots, N\}$ that maps past feedback sequences to actions. That the algorithms are deterministic is assumed for convenience. In particular, the lower bounds we prove can be extended to randomized algorithms by conditioning on the internal randomization of the algorithm. Note that the algorithms we design are themselves deterministic.

6.2 Classification of finite partial-monitoring games

In this section we present the theorem that classifies all finite stochastic partial-monitoring games based on how their minimax regret scales with the time horizon along with an illustration and the proof of the theorem.

Theorem 9 (Classification)**.** *Let* $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ *be a partial-monitoring game with* N *actions and* M *outcomes. Let* $\mathcal{C} = \{C_1, \ldots, C_k\}$ *be its cell decomposition, with corresponding loss vectors* $\ell_1, \ldots, \ell_k$. *The game* $\mathbf{G}$ *falls into one of the following four categories:*

(a) $R_T(\mathbf{G}) = 0$ *if there exists an action* i *with* $C_i = \Delta_M$. *This case is called* trivial.

(b) $R_T(\mathbf{G}) = \Theta(T)$ *if there exist two Pareto-optimal actions* i *and* j *such that* $\ell_i - \ell_j$ *is* not *globally observable. This case is called* hopeless.

(c) $R_T(\mathbf{G}) = \widetilde{\Theta}(\sqrt{T})$ *if it is not trivial and it satisfies the local observability condition. These games are called* easy.

(d) $R_T(\mathbf{G}) = \Theta(T^{2/3})$ *if* $\mathbf{G}$ *is not hopeless and it does not satisfy the local observability condition. These games are called* hard.

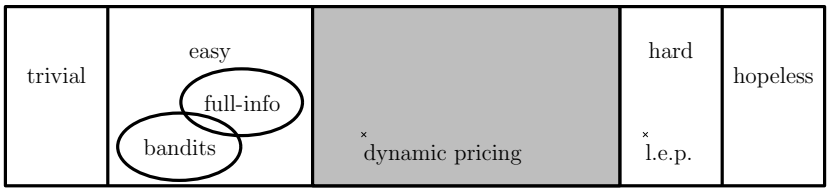


Figure 6.1: Partial monitoring games and their minimax regret as it was known previously. The big rectangle denotes the set of all games. Inside the big rectangle, the games are ordered from left to right based on their minimax regret. In the "hard" area, l.e.p. denotes label-efficient prediction. The grey area contains games whose minimax regret is between $\Omega(\sqrt{T})$ and $O(T^{2/3})$ but their exact regret rate was unknown. This area is now eliminated, and the dynamic pricing problem is proven to be hard.

Note that the conditions listed under (a)–(d) are mutually exclusive and cover all finite partial-monitoring games. The only non-obvious implication is that if a game is easy then it cannot be hopeless. The reason this holds is because for any pair of cells C_i, C_j in $\mathcal{C}$, the vector $\ell_i - \ell_j$ can be expressed as a telescoping sum of the differences of loss vectors of neighboring cells.

The next sections are dedicated to proving Theorem 9. We start with the simple cases. If there exists an action whose cell covers the whole probability simplex then choosing that action in every round will yield zero regret, proving case (a). The condition in Case (b) is due to Piccolboni and Schindelhauer [2001], who showed that under the condition mentioned there, there is no algorithm that achieves sublinear regret[3]. The upper bound for case (d) is achieved by the FeedExp3 algorithm due to Piccolboni and Schindelhauer [2001], for which a regret bound of $O(T^{2/3})$ was shown by Cesa-Bianchi et al. [2006]. The lower bound for case (c) can be found in Chapter 4 (Theorem 4). For a visualization of previous results, see Figure 6.1.

The above assertions help characterize trivial and hopeless games, and show that if a game is not trivial and not hopeless then its minimax regret falls between $\Omega(\sqrt{T})$ and $O(T^{2/3})$. Our contribution in this chapter is that we give exact minimax rates (up to logarithmic factors) for these games. To prove the upper bound for case (c), we introduce a new algorithm, which we call Balaton, for "Bandit Algorithm for Loss Annihilation"[4]. This algorithm is presented in Section 6.2.1, while its analysis is given in Section 6.2.2. The lower bound for case (d) is presented in Section 6.2.3.

[3]Although Piccolboni and Schindelhauer state their theorem for adversarial environments, their proof applies to stochastic environments without any change (which is important for the lower bound part).

[4]Balaton is also the name of a lake in Hungary. We thank Gergely Neu for suggesting the name.

Example

Before getting into proving Theorem 9, we demonstrate its strength with the help of an example. Namely, we show that the discretized dynamic pricing game (see Example 6 in Section 1.3) is *hard*. Recall that dynamic pricing is a game between a vendor (learner) and a customer (environment), where in each round, the vendor sets a price he wants to sell his product at (action), and the costumer sets a maximum price he is willing to buy the product for (outcome). If the product is not sold, the vendor suffers some constant loss, otherwise his loss is the difference between the customer's maximum and his price. The customer never reveals the maximum price and thus the vendor's only feedback is whether he sold the product or not.

The discretized version of the game with N actions (and outcomes) is defined by the matrices

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 2 & \cdots & N-1 \\ c & 0 & 1 & \cdots & N-2 \\ \vdots & & \ddots & & \vdots \\ c & \cdots & c & 0 & 1 \\ c & \cdots & \cdots & c & 0 \end{pmatrix} \qquad \mathbf{H} = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix},$$

where c is a positive constant (see Figure 6.2 for the cell-decomposition for $N = 3$). It is easy to see that all the actions are Pareto-optimal. Also, after some linear algebra it turns out that the point

$$p^* = \left(\frac{1}{1+c}, \frac{c}{(1+c)^2}, \frac{c^2}{(1+c)^3}, \ldots, \frac{c^{N-2}}{(1+c)^{N-1}}, \frac{c^{N-1}}{(1+c)^{N-1}} \right)^\top$$

is a common vertex of all cells in the interior of the probability simplex. It follows that any two actions are neighbors. On the other hand, if we take two actions i and i' such that $|i - i'| \neq 1$, $\ell_i - \ell_{i'}$ is not locally observable. For example, the signal matrices for actions 1 and action N are

$$S_1 = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} \qquad S_N = \begin{pmatrix} 1 & \cdots & 1 & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix},$$

whereas $\ell_N - \ell_1 = (c, c-1, \ldots, c-N+2, -N+1)^\top$. It is obvious that $\ell_N - \ell_1$ is not in the direct sum of the row spaces of S_1 and S_N, and thus by Theorem 9, the game of dynamic pricing is hard.

6.2.1 Balaton: An algorithm for easy games

In this section we present our algorithm that achieves $\widetilde{O}(\sqrt{T})$ expected regret for easy games (case (c) of Theorem 9). The input of the algorithm is the loss matrix $\mathbf{L}$, the feedback matrix $\mathbf{H}$, the time horizon T and an error probability δ, to be chosen later. Before describing the algorithm, we introduce some

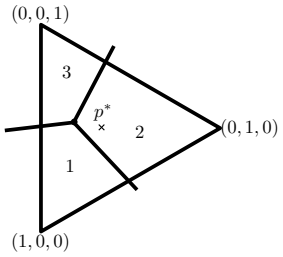


Figure 6.2: The cell decomposition of the discretized dynamic pricing game with 3 actions. If the opponent strategy is p^*, then action 2 is the optimal action.

notation. We define a graph $\mathcal{G}$ associated with game $\mathbf{G}$ the following way. Let the vertex set be the set of cells of the cell decomposition $\mathcal{C}$ of the probability simplex such that cells $C_i, C_j \in \mathcal{C}$ share the same vertex when $C_i = C_j$. The graph has an edge between vertices whose corresponding cells are neighbors. This graph is connected, since the probability simplex is convex and the cell decomposition covers the simplex.

Recall that for neighboring actions i, j, if the game is locally observable, $\ell_i - \ell_j \in \oplus_{k \in N_{i,j}^+} \operatorname{Im} S_k^\top$. It follows that there exist *observer vectors* $v_{i,j,k}$ such that $\ell_i - \ell_j = \sum_{k \in N_{i,j}^+} S_k^\top v_{i,j,k}$. The existence of these vectors is the core property of locally observable games that makes it possible to have $\widetilde{O}(\sqrt{T})$ regret upper bound because, as we will see, it enables us to estimate loss differences of neighboring actions for a low cost.

The main idea of the algorithm is to successively eliminate actions in an efficient, yet safe manner. When all remaining Pareto-optimal actions share the same cell, the elimination phase finishes and from this point, one of the remaining actions is played. During the elimination phase, the algorithm works in rounds. In each round every "alive" action is played once. The resulting observations are used to estimate the loss-difference between the alive actions. If some estimate becomes sufficiently precise, the action of the pair deemed to be suboptimal is eliminated (possibly together with some other actions). To determine if an estimate is sufficiently precise, we will use an appropriate stopping rule. A small regret will be achieved by tuning the error probability of the stopping rule appropriately.

The details of the algorithm are as follows: In the preprocessing phase, the algorithm constructs the neigbourhood graph, the neighorhood action sets $N_{i,j}^+$ assigned to the edges of the graph, the signal matrices S_i, and the vectors $v_{i,j,k}$. In addition, it constructs a path in the graph connecting any pairs of nodes, and initializes some variables used by the stopping rule.

In the elimination phase, the algorithm runs a loop. In each round of the loop, the algorithm chooses each of the alive actions once and, based on the observations, the estimates $\hat{\delta}_{i,j}$ of the loss-differences $\delta_{i,j} \stackrel{\triangle}{=} (\ell_i - \ell_j)^\top p^*$ are updated, where p^* is the actual opponent strategy. The algorithm maintains

Algorithm 14 BALATON

Input: $\mathbf{L}, \mathbf{H}, T, \delta$
Initialization:
$[\mathcal{G}, \mathcal{C}, \{v_{i,j,k}\}, \{path_{(i,j)}\}, \{(LB_{(i,j)}, UB_{(i,j)}, \sigma_{(i,j)}, R_{(i,j)})\}] \leftarrow$INITIALIZE$(\mathbf{L}, \mathbf{H})$
$t \leftarrow 0$, $n \leftarrow 0$
$aliveActions \leftarrow \{1 \le i \le N \ : \ C_i \cap interior(\Delta_M) \neq \emptyset\}$
main loop
while $|V_{\mathcal{G}}| > 1$ and $t < T$ **do**
  $n \leftarrow n + 1$
  for each $i \in aliveActions$ **do**
    $O_i \leftarrow$ EXECUTEACTION(i)
    $t \leftarrow t + 1$
  end for
  for each edge (i, j) in $\mathcal{G}$: $\hat{\delta}_{i,j} \leftarrow \sum_{k \in N^+_{i,j}} Y_k^\top v_{i,j,k}$ **end for**
  for each non-adjacent pair (i, j) in $\mathcal{G}$: $\hat{\delta}_{i,j} \leftarrow \sum_{(k,l) \in path_{(i,j)}} \hat{\delta}_{k,l}$ **end for**
  $haveEliminated \leftarrow$ false
  for each vertex pair (i, j) in $\mathcal{G}$ **do**
    $\tilde{\delta}_{i,j} \leftarrow \left(1 - \frac{1}{n}\right) \tilde{\delta}_{i,j} + \frac{1}{n} \hat{\delta}_{i,j}$
    if BSTOPSTEP$(\tilde{\delta}_{i,j}, LB_{(i,j)}, UB_{(i,j)}, \sigma_{(i,j)}, R_{(i,j)}, n, 1/2, \delta)$ **then**
      $toEliminate(i, j) \leftarrow \mathrm{sgn}(\tilde{\delta}_{i,j})$
      $haveEliminated \leftarrow$ true
    else
      $toEliminate(i, j) \leftarrow 0$
    end if
  end for
  $[aliveActions, \mathcal{C}, \mathcal{G}] \leftarrow$ELIMINATE$(toEliminate)$
  if $haveEliminated$ **then**
    $\{path_{(i,j)}\} \leftarrow$REGENERATEPATHS$(\mathcal{G})$
  end if
end while
Let i be a Pareto-optimal action in $aliveActions$
while $t < T$ **do**
  EXECUTEACTION(i)
  $t \leftarrow t + 1$
end while

the set $\mathcal{C}$ of cells of alive actions and their neighborhood graph $\mathcal{G}$.

The estimates are calculated as follows. First we calculate estimates for neighboring actions (i, j). In round[5] n, for every action k in $N^+_{i,j}$ let Y_k be the observation vector for action k. Let $\hat{\delta}_{i,j} = \sum_{k \in N^+_{i,j}} Y_k^\top v_{i,j,k}$. From the local observability condition and the construction of $v_{i,j,k}$, with simple algebra it

[5]Note that a round of the algorithm is not the same as the time step t. In a round, the algorithm chooses each of the alive actions once.

follows that $\hat{\delta}_{i,j}$ are unbiased estimates of $(\ell_i - \ell_j)^\top p^*$ (see Lemma 14). For non-neighboring action pairs, we use telescoping sums: since the graph $\mathcal{G}$ (induced by the alive actions) stays connected, we can take a path $i = i_0, i_1, \ldots, i_r = j$ in the graph, and the estimate $\hat{\delta}_{i,j}$ will be the sum of the estimates along the path: $\sum_{l=1}^{r} \hat{\delta}_{i_{l-1},i_l}$. The estimate of the difference of the expected losses after round n will be the average $\tilde{\delta}_{i,j} = (1/n) \sum_{l=1}^{n} \hat{\delta}_{i,j}(s)$, where $\hat{\delta}_{i,j}(s)$ denotes the estimate for pair (i, j) computed in round s.

After updating the estimates, the algorithm decides which actions to eliminate. For each pair of vertices i, j of the graph, the expected difference of their loss is tested for its sign by the BStopStep subroutine, based on the estimate $\tilde{\delta}_{i,j}$ and its relative error. This subroutine uses a stopping rule based on Bernstein's inequality.

The subroutine's pseudocode is shown as Algorithm 15 and is essentially based on the work by Mnih et al. [2008]. The algorithm maintains two values, LB, UB, computed from the supplied sequence of sample means ($\hat{\mu}$) and the deviation bounds

$$c(\sigma, R, n, \delta) = \sigma \sqrt{\frac{2\, L(\delta, n)}{n}} + \frac{R\, L(\delta, n)}{3n}, \text{ where } L(\delta, n) = \log\left(3\, \frac{p}{p-1}\, \frac{n^p}{\delta}\right). \tag{6.1}$$

Here $p > 1$ is an arbitrarily chosen parameter of the algorithm, σ is a (deterministic) upper bound on the (conditional) variance of the random variables whose common mean μ we wish to estimate, while R is a (deterministic) upper bound on their range. This is a general stopping rule method, which stops when it produced an ϵ-relative accurate estimate of the unknown mean. The algorithm is guaranteed to be correct outside of a failure event whose probability is bounded by δ.

Algorithm Balaton calls this method with $\varepsilon = 1/2$. As a result, when BStopStep returns true, outside of the failure event the sign of the estimate $\tilde{\delta}$ supplied to Balaton will match the sign of the mean to be estimated. The conditions under which the algorithm indeed produces ε-accurate estimates (with high probability) are given in Lemma 28 (see Appendix), which also states that also with high probability, the time when the algorithm stops is bounded by

$$C \cdot \max\left(\frac{\sigma^2}{\epsilon^2 \mu^2}, \frac{R}{\epsilon|\mu|}\right)\left(\log\frac{1}{\delta} + \log\frac{R}{\epsilon|\mu|}\right),$$

where $\mu \neq 0$ is the true mean. Note that the choice of p in (6.1) influences only C.

If BStopStep returns true for an estimate $\tilde{\delta}_{i,j}$, then the i, j pair becomes a candidate for elimination. After checking all pairs, function eliminate is called. If, say, $\tilde{\delta}_{i,j} > 0$, this function takes the closed half space $\{q \in \Delta_M : (\ell_i - \ell_j)^\top q \leq 0\}$ and eliminates *all* actions whose cell lies completely in the half space. The function also drops the vertices from the graph that correspond to eliminated cells. The elimination necessarily concerns all actions

Algorithm 15 Algorithm BStopStep. Note that, somewhat unusually at least in pseudocodes, the arguments LB, UB are passed by reference, i.e., the algorithm rewrites the values of these arguments (which are thus returned back to the caller).

Input: $\hat{\mu}, \mathrm{LB}, \mathrm{UB}, \sigma, R, n, \varepsilon, \delta$
$\mathrm{LB} \leftarrow \max(\mathrm{LB}, |\hat{\mu}| - c(\delta, \sigma, R, n))$
$\mathrm{UB} \leftarrow \min(\mathrm{UB}, |\hat{\mu}| + c(\delta, \sigma, R, n))$
Return $(1+\epsilon)\mathrm{LB} < (1-\epsilon)\mathrm{UB}$

with corresponding cell C_i, and possibly other actions as well. The remaining cells are redefined by taking their intersection with the complement half space $\{q \in \Delta_M : (\ell_i - \ell_j)^\top q \geq 0\}$.

By construction, after the elimination phase, the remaining graph is still connected, but some paths used in the round may have lost vertices or edges. For this reason, in the last phase of the round, new paths are constructed for vertex pairs with broken paths.

The main loop of the algorithm continues until either one vertex remains in the graph or the time horizon T is reached. In the former case, one of the actions corresponding to that vertex is chosen until the time horizon is reached.

6.2.2 Analysis of the algorithm

In this section we prove that the algorithm described in the previous section achieves $\widetilde{O}(\sqrt{T})$ expected regret.

Theorem 10. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be a finite partial-monitoring game that satisfies the local observability condition. Then there exists a game-dependent constant C such that for every time horizon T, with appropriately tuned parameter δ, the expected regret of* Balaton *is upper bounded as*

$$\mathbb{E}[R_T] \leq C\sqrt{T}\log T\,.$$

Let us assume that the outcomes are generated following the probability vector $p^* \in \Delta_M$. Let j^* denote an optimal action, that is, for every $1 \leq i \leq N$, $\ell_{j^*}^\top p^* \leq \ell_i^\top p^*$. For every pair of actions i, j, let $\delta_{i,j} = (\ell_i - \ell_j)^\top p^*$ be the expected difference of their instantaneous loss. The expected regret of the algorithm can be rewritten as

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbf{L}[I_t, J_t] - \min_{1\leq i\leq N} \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{L}[i, J_t]\right]\right] = \sum_{i=1}^{N} \mathbb{E}\left[\tau_i\right] \delta_{i,j^*}\,, \tag{6.2}$$

where τ_i is the number of times action i is chosen by the algorithm.

Throughout the proof, the value that Balaton assigns to a variable x in round n will be denoted by $x(n)$. Further, for $1 \leq k \leq N$, we introduce the

i.i.d. random sequence $(J_k(n))_{n\geq 1}$, taking values on $\{1,\dots,M\}$, with common multinomial distribution satisfying, $\mathbb{P}\left[J_k(n)=j\right]=p_j^*$. Clearly, a statistically equivalent model to the one where (J_t) is an i.i.d. sequence with multinomial p^* is when (J_t) is defined through

$$J_t = J_{I_t}\left(\sum_{s=1}^{t}\mathbb{I}_{\{I_s=I_t\}}\right)\,. \tag{6.3}$$

Note that this claim holds, independently of the algorithm generating the actions, I_t. Therefore, in what follows, we assume that the outcome sequence is generated through (6.3). As we will see, this construction significantly simplifies subsequent steps of the proof. If action k is selected by our algorithm in the n^{th} elimination round, then the outcome obtained in response is going to be $Y_k(n)=S_k u_k(n)$, where $u_k(n)=e_{J_k(n)}$. (This holds because in the elimination rounds all alive actions are tried exactly once by Balaton.)

Let $(\mathcal{F}_n)_n$ be the filtration defined as $\mathcal{F}_n=\sigma(u_k(m); 1\leq k\leq N, 1\leq m\leq n)$. We also introduce the notations $\mathbb{E}_n[\cdot]=\mathbb{E}[\cdot|\mathcal{F}_n]$ and $\mathbb{V}\mathrm{ar}_n(\cdot)=\mathbb{V}\mathrm{ar}(\cdot|\mathcal{F}_n)$, the conditional expectation and conditional variance operators corresponding to $\mathcal{F}_n$. Note that $\mathcal{F}_n$ contains the information known to Balaton (and more) at the end of the elimination round n. Our first (trivial) observation is that $\hat{\delta}_{i,j}(n)$, the estimate of $\delta_{i,j}$ obtained in round n is $\mathcal{F}_n$-measurable. The next lemma establishes that, furthermore, $\hat{\delta}_{i,j}(n)$ is an unbiased estimate of $\delta_{i,j}$:

Lemma 14. *For any $n\geq 1$ and i,j such that $C_i,C_j\in\mathcal{C}$, $\mathbb{E}_{n-1}[\hat{\delta}_{i,j}(n)]=\delta_{i,j}$.*

The following lemma upper bounds the conditional variance and the range of the estimates.

Lemma 15. *The conditional variance of $\hat{\delta}_{i,j}(n)$, $\mathbb{V}\mathrm{ar}_{n-1}(\hat{\delta}_{i,j}(n))$, is upper bounded by $V=2\sum_{\{i,j\in\mathcal{L}\}}\sum_{k\in N_{i,j}^+}\|v_{i,j,k}\|_2^2$. The range of the estimates $\hat{\delta}_{i,j}(n)$ is upper bounded by $R=\sum_{\{i,j\}\in\mathcal{L}}\sum_{k\in N_{i,j}^+}\|v_{i,j,k}\|_1$.*

Let δ be the confidence parameter used in BStopStep. Since, according to Lemmas 14 and 15, $(\hat{\delta}_{i,j})$ is a "shifted" martingale difference sequence with conditional mean $\delta_{i,j}$, bounded conditional variance and range, we can apply Lemma 28 stated in the Appendix. By the union bound, the probability that any of the confidence bounds fails during the game is at most $N^2\delta$. Thus, with probability at least $1-N^2\delta$, if BStopStep returns true for a pair (i,j) then $\mathrm{sgn}(\delta_{i,j})=\mathrm{sgn}(\hat{\delta}_{i,j})$ and the algorithm eliminates all the actions whose cell is contained in the closed half space defined by $\mathcal{H}=\{p\,:\,\mathrm{sgn}(\delta_{i,j})p^\top(\ell_i-\ell_j)\leq 0\}$. By definition $\delta_{i,j}=(\ell_i-\ell_j)^\top p^*$. Thus $p^*\notin\mathcal{H}$ and none of the eliminated actions can be optimal under p^*.

From Lemma 28 we also see that, with probability at least $1-N^2\delta$, the number of times τ_i^* the algorithm experiments with a suboptimal action i during the elimination phase is bounded by

$$\tau_i^*\leq\frac{c(\mathbf{G})}{\delta_{i,j^*}^2}\log\frac{R}{\delta\delta_{i,j^*}}=T_i\,, \tag{6.4}$$

where $c(\mathbf{G}) = C(V+R)$ is a problem dependent constant.

The following lemma shows that degenerate actions will be eliminated in time.

Lemma 16. *Let action i be a degenerate action. Let $N_i^+ = \{j \ : \ C_j \in \mathcal{C}, C_i \subset C_j\}$. The following two statements hold:*

1. *If any of the actions in N_i^+ is eliminated, then action i is eliminated as well.*

2. *There exists an action $k_i \in N_i^+$ such that $\delta_{k_i,j^*} \geq \delta_{i,j^*}$.*

An immediate implication of the first claim of the lemma is that if action k_i gets eliminated then action i gets eliminated as well, that is, the number of times action i is chosen cannot be greater than that of action k_i. Hence, $\tau_i^* \leq \tau_{k_i}^*$.

Let $\mathcal{E}$ be the failure event underlying the stopping rules. As discussed earlier, $\mathbb{P}(\mathcal{E}) \leq N^2\delta$. Note that on $\mathcal{E}^c$, i.e., when the stopping rules do not fail, no suboptimal action can remain for the final phase. Hence, $\tau_i \mathbb{I}_{\{\mathcal{E}^c\}} \leq \tau_i^* \mathbb{I}_{\{\mathcal{E}^c\}}$, where τ_i is the number of times action i is chosen by the algorithm. To upper bound the expected regret we continue from (6.2) as

$$
\begin{aligned}
&\sum_{i=1}^{N} \mathbb{E}\left[\tau_i\right] \delta_{i,j^*} \\
&= \sum_{i=1}^{N} \mathbb{E}\left[\mathbb{I}_{\{\mathcal{E}^c\}}\tau_i\right] \delta_{i,j^*} + \mathbb{P}(\mathcal{E})\, T \ (\text{because} \textstyle\sum_{i=1}^{N} \tau_i = T \ , \ 0 \leq \delta_{i,j^*} \leq 1) \\
&\leq \sum_{i=1}^{N} \mathbb{E}\left[\mathbb{I}_{\{\mathcal{E}^c\}}\tau_i^*\right] \delta_{i,j^*} + N^2\delta T \\
&\leq \sum_{i:\ C_i \in \mathcal{C}} \mathbb{E}\left[\mathbb{I}_{\{\mathcal{E}^c\}}\tau_i^*\right] \delta_{i,j^*} + \sum_{i:\ C_i \notin \mathcal{C}} \mathbb{E}\left[\mathbb{I}_{\{\mathcal{E}^c\}}\tau_i^*\right] \delta_{i,j^*} + N^2\delta T \\
&\leq \sum_{i:\ C_i \in \mathcal{C}} \mathbb{E}\left[\mathbb{I}_{\{\mathcal{E}^c\}}\tau_i^*\right] \delta_{i,j^*} + \sum_{i:\ C_i \notin \mathcal{C}} \mathbb{E}\left[\mathbb{I}_{\{\mathcal{E}^c\}}\tau_{k_i}^*\right] \delta_{k_i,j^*} + N^2\delta T \ (\text{by Lemma 16}) \\
&\leq \sum_{i:\ C_i \in \mathcal{C}} T_i \delta_{i,j^*} + \sum_{i:\ C_i \notin \mathcal{C}} T_{k_i} \delta_{k_i,j^*} + N^2\delta T \\
&\leq \sum_{\substack{i:\ C_i \in \mathcal{C} \\ \delta_{i,j^*} \geq \delta_0}} T_i \delta_{i,j^*} + \sum_{\substack{i:\ C_i \notin \mathcal{C} \\ \delta_{k_i,j^*} \geq \delta_0}} T_{k_i} \delta_{k_i,j^*} + \left(\delta_0 + N^2\delta\right) T \\
&\leq c(\mathbf{G}) \left(\sum_{\substack{i:\ C_i \in \mathcal{C} \\ \delta_{i,j^*} \geq \delta_0}} \frac{\log \frac{R}{\delta\delta_{i,j^*}}}{\delta_{i,j^*}} + \sum_{\substack{i:\ C_i \notin \mathcal{C} \\ \delta_{k_i,j^*} \geq \delta_0}} \frac{\log \frac{R}{\delta\delta_{k_i,j^*}}}{\delta_{k_i,j^*}} \right) + \left(\delta_0 + N^2\delta\right) T \\
&\leq c(\mathbf{G}) N \frac{\log \frac{R}{\delta\delta_0}}{\delta_0} + \left(\delta_0 + N^2\delta\right) T,
\end{aligned}
$$

The above calculation holds for any value of $\delta_0 > 0$. Setting

$$\delta_0 = \sqrt{\frac{c(\mathbf{G})N}{T}} \qquad \text{and} \qquad \delta = \sqrt{\frac{c(\mathbf{G})}{TN^3}}\,, \qquad \text{we get}$$

$$\mathbb{E}\left[R_T\right] \leq \sqrt{c(\mathbf{G})NT}\log\left(\frac{RTN^2}{c(\mathbf{G})}\right).$$

In conclusion, if we run BALATON with parameter $\delta = \sqrt{\frac{c(\mathbf{G})}{TN^3}}$, the algorithm suffers regret of $\widetilde{O}(\sqrt{T})$, finishing the proof.

6.2.3 A lower bound for hard games

In this section we prove that for any game that satisfies the condition of Case (d) of Theorem 9, the minimax regret is of $\Omega(T^{2/3})$.

Theorem 11. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be an N by M partial-monitoring game. Assume that there exist two neighboring actions i and j that do not satisfy the local observability condition. Then there exists a problem dependent constant $c(\mathbf{G})$ such that for any algorithm $\mathcal{A}$ and time horizon T there exists an opponent strategy p such that the expected regret satisfies*

$$\mathbb{E}\left[R_T^{\mathcal{A}}(p)\right] \geq c(\mathbf{G})T^{2/3}\,.$$

Proof. Without loss of generality, we can assume that the two neighbor cells in the condition are C_1 and C_2. Let $C_3 = C_1 \cap C_2$. For $i = 1, 2, 3$, let A_i be the set of actions associated with cell C_i. Note that A_3 may be the empty set. Let $A_4 = A \setminus (A_1 \cup A_2 \cup A_3)$. By our convention for naming loss vectors, ℓ_1 and ℓ_2 are the loss vectors for C_1 and C_2, respectively. Let Λ_3 collect the loss vectors of actions that lie on the open segment connecting ℓ_1 and ℓ_2. It is easy to see that Λ_3 is the set of loss vectors that correspond to the cell C_3. We define Λ_4 as the set of all the other loss vectors. For $i = 1, 2, 3, 4$, let $k_i = |A_i|$.

Since actions 1 and 2 are not locally observable neighbors, $\ell_2 - \ell_1 \notin \bigoplus_{i \in A_1 \cup A_2 \cup A_3} \operatorname{Im} S_i^\top$. It follows from the De Morgan rule and from the fact that for any matrix M, $(\operatorname{Im} M)^\perp = \operatorname{Ker}(M^\top)$, that $(\ell_2 - \ell_1)^\perp \not\supset \bigcap_{i \in A_1 \cup A_2 \cup A_3} \operatorname{Ker} S_i$. Thus, there exists a vector v such that for all $i \in A_1 \cup A_2 \cup A_3$, $v \in \operatorname{Ker} S_i$ and also $(\ell_2 - \ell_1)^\top v \neq 0$. By scaling we can assume that $(\ell_2 - \ell_1)^\top v = 1$. Note that since the rowspace of any signal matrix S_i always contains the vector $(1, 1, \ldots, 1)$, the coordinates of v sum up to zero.

Let p_0 be an arbitrary probability vector in the relative interior of C_3. It is easy to see that for any $\varepsilon > 0$ small enough, $p_1 = p_0 + \varepsilon v \in C_1 \setminus C_2$ and $p_2 = p_0 - \varepsilon v \in C_2 \setminus C_1$.

Let us fix a deterministic algorithm $\mathcal{A}$ and a time horizon T. For $i = 1, 2$, let $R_T^{(i)}$ denote the expected regret of the algorithm under opponent strategy p_i. For $i = 1, 2$ and $j = 1, \ldots, 4$, let N_j^i denote the expected number of times the

algorithm chooses an action from A_j, assuming the opponent plays strategy p_i.

From the definition of Λ_3 we know that for any $\ell \in \Lambda_3$, $\ell - \ell_1 = \eta_\ell(\ell_2 - \ell_1)$ and $\ell - \ell_2 = (1-\eta_\ell)(\ell_1 - \ell_2)$ for some $0 < \eta_\ell < 1$. Let $\lambda_1 = \min_{\ell\in\Lambda_3} \eta_\ell$ and $\lambda_2 = \min_{\ell\in\Lambda_3}(1-\eta_\ell)$ and $\lambda = \min(\lambda_1, \lambda_2)$ if $\Lambda_3 \neq \emptyset$ and let $\lambda = 1/2$, otherwise. Finally, let $\beta_i = \min_{\ell\in\Lambda_4}(\ell - \ell_i)^\top p_i$ and $\beta = \min(\beta_1, \beta_2)$. Note that $\lambda, \beta > 0$.

As the first step of the proof, we lower bound the expected regret $R_T^{(1)}$ and $R_T^{(2)}$ in terms of the values $N_j^i, \varepsilon, \lambda$ and β:

$$\begin{aligned} R_T^{(1)} &\geq N_2^1 \overbrace{(\ell_2 - \ell_1)^\top p_1}^{\varepsilon} + N_3^1 \lambda (\ell_2 - \ell_1)^\top p_1 + N_4^1 \beta \geq \lambda(N_2^1 + N_3^1)\varepsilon + N_4^1 \beta\,, \\ R_T^{(2)} &\geq N_1^2 \underbrace{(\ell_1 - \ell_2)^\top p_2}_{\varepsilon} + N_3^2 \lambda (\ell_1 - \ell_2)^\top p_2 + N_4^2 \beta \geq \lambda(N_1^2 + N_3^2)\varepsilon + N_4^2 \beta\,. \end{aligned} \tag{6.5}$$

For the next step, we need the following lemma.

Lemma 17. *There exists a (problem dependent) constant c such that for any small enough ε, the following inequalities hold:*

$$\begin{aligned} N_1^2 &\geq N_1^1 - cT\varepsilon\sqrt{N_4^1}\,, & N_3^2 &\geq N_3^1 - cT\varepsilon\sqrt{N_4^1}\,, \\ N_2^1 &\geq N_2^2 - cT\varepsilon\sqrt{N_4^2}\,, & N_3^1 &\geq N_3^2 - cT\varepsilon\sqrt{N_4^2}\,. \end{aligned}$$

Now we can continue lower bounding the expected regret.

Let $r = \operatorname{argmin}_{i\in\{1,2\}} N_4^i$. It is easy to see that for $i = 1, 2$ and $j = 1, 2, 3$,

$$N_j^i \geq N_j^r - c_2 T\varepsilon\sqrt{N_4^r}\,.$$

If $i \neq r$ then this inequality is one of the inequalities from Lemma 17. If $i = r$ then it is a trivial lower bounding by subtracting a positive value. From (6.5) we have

$$\begin{aligned} R_T^{(i)} &\geq \lambda(N_{3-i}^i + N_3^i)\varepsilon + N_4^i\beta \\ &\geq \lambda(N_{3-i}^r - c_2 T\varepsilon\sqrt{N_4^r} + N_3^r - c_2 T\varepsilon\sqrt{N_4^r})\varepsilon + N_4^r\beta \\ &= \lambda(N_{3-i}^r + N_3^r - 2c_2 T\varepsilon\sqrt{N_4^r})\varepsilon + N_4^r\beta\,. \end{aligned}$$

Now assume that, at the beginning of the game, the opponent randomly chooses between strategies p_1 and p_2 with equal probability. Then the expected regret of the algorithm is lower bounded by

$$\begin{aligned} R_T &= \frac{1}{2}\left(R_T^{(1)} + R_T^{(2)}\right) \\ &\geq \frac{1}{2}\lambda(N_1^r + N_2^r + 2N_3^r - 4c_2 T\varepsilon\sqrt{N_4^r})\varepsilon + N_4^r\beta \\ &\geq \frac{1}{2}\lambda(N_1^r + N_2^r + N_3^r - 4c_2 T\varepsilon\sqrt{N_4^r})\varepsilon + N_4^r\beta \\ &= \frac{1}{2}\lambda(T - N_4^r - 4c_2 T\varepsilon\sqrt{N_4^r})\varepsilon + N_4^r\beta\,. \end{aligned}$$

Choosing $\varepsilon = c_3 T^{-1/3}$ we get

$$\begin{aligned} R_T &\geq \frac{1}{2}\lambda c_3 T^{2/3} - \frac{1}{2}\lambda N_4^r c_3 T^{-1/3} - 2\lambda c_2 c_3^2 T^{1/3}\sqrt{N_4^r} + N_4^r \beta \\ &\geq T^{2/3}\left(\left(\beta - \frac{1}{2}\lambda c_3\right)\frac{N_4^r}{T^{2/3}} - 2\lambda c_2 c_3^2\sqrt{\frac{N_4^r}{T^{2/3}}} + \frac{1}{2}\lambda c_3\right) \\ &= T^{2/3}\left(\left(\beta - \frac{1}{2}\lambda c_3\right)x^2 - 2\lambda c_2 c_3^2 x + \frac{1}{2}\lambda c_3\right), \end{aligned}$$

where $x = \sqrt{N_4^r/T^{2/3}}$. Now we see that $c_3 > 0$ can be chosen to be small enough, independently of T so that, for any choice of x, the quadratic expression in the parenthesis is bounded away from zero, and simultaneously, ε is small enough so that the threshold condition in Lemma 17 is satisfied, completing the proof of Theorem 11. □

6.2.4 Summary

The previous sections were devoted to proving the classification theorem (Theorem 9). For the upper bound on the minimax regret of easy games, the algorithm Balaton was introduced in Section 6.2.1, and an upper bound in its expected regret was proven in Section 6.2.2. Next, a lower bound on the minimax regret of hard games was shown in Section 6.2.3, completing the proof of the theorem.

As seen from Theorem 9, the crucial condition that separates easy games from hard games is the local observability condition (Definition 10). It is important to note that the classification theorem in this chapter only deals with games with stochastic opponents. However, we conjectured that the theorem remains true if we lift this assumption. That is, games with adversarial opponents have the same classification, with the same condition separating easy and hard games:

Conjecture 1. *Any N by M partial-monitoring game against adversarial opponents can be classified into four categories based on the growth rate of its minimax regret in the following way:*

1. *The game is* trivial *and has* 0 *minimax regret if in its cell decomposition, there exists a cell $C_i = \Delta_M$.*

2. *The game is* easy *with minimax regret $\widetilde{\Theta}(\sqrt{T})$ if it is not trivial and it satisfies the local observability condition.*

3. *The game is* hard *with minimax regret $\Theta(T^{2/3})$ if it satisfies the global observability condition but does not satisfy the local observability condition.*

4. *The game is* hopeless *with minimax regert $\Theta(T)$ is it does not satisfy the global observability condition.*

Since FeedExp3 of Piccolboni and Schindelhauer [2001] achieves $O(T^{2/3})$ regret against any non-hopeless game with adversarial opponent and all the lower bounds in this chapter hold for adversarial opponents as well, the only part left to prove the above conjecture is to design an algorithm that achieves $\widetilde{O}(\sqrt{T})$ regret on easy games against non-stochastic opponents. This was partially done by Foster and Rakhlin [2011], who designed the algorithm NeighborhoodWatch. This algorithm achieves $O(\sqrt{T})$ regret for *non-degenerate* games with local observability. We summarize their result in the next section. If the conjecture is true for degenerate games remains an open problem.

6.2.5 NeighborhoodWatch: an algorithm against non-stochastic environments by Foster and Rakhlin [2011]

In their paper, Foster and Rakhlin introduce the algorithm NeighborhoodWatch and show that it achieves $O(\sqrt{T})$ minimax regret on non-degenerate finite partial-monitoring games against non-stochastic environments. Here the term non-degenerate means that there are no degenerate actions as well as no "duplicate" actions, that is, actions whose cells $C_i = C_j$. Without loss of generality it is also assumed that there are no dominated actions (to recall the definition of degenerate and dominated actions refer to Definition 7).

The algorithm works as follows. At the beginning, the game is split to "local" games: for every action i, the local game N_i associated with i is defined as the action i and all of its neighbors. For each of these local games, an internal algorithm $\mathcal{A}_i$ is assigned. Algorithm $\mathcal{A}_i$ plays on the local game N_i and at every time step when revoked, chooses an action based on a probability vector $q_t^i \in \mathbb{R}^N$, where the coordinates of q_t^i associated to actions not in N_i are zero. The question of which local game to revoke at time step t is decided by a meta-algorithm.

The meta-algorithm chooses a local game randomly, and the distribution based on which the local game is chosen is defined the following way. At time step t, the matrix $Q_t \in \mathbb{R}^{N\times N}$ is defined as

$$Q_t = \begin{pmatrix} q_t^1 & q_t^2 & \cdots & q_t^N \end{pmatrix} .$$

Then, the probability vector $p_t \in \mathbb{R}^N$ is defined as a *fixed-point* of Q_t:

$$p_t = Q_t p_t .$$

The advantage of this setup is manifold. First of all, due to the construction of the local games, it is possible to construct unbiased estimates of the loss differences of action i and its neighbors in the local game N_i, exploiting the local observability condition. Second, it can be shown that if action i happens to be optimal then the "second best" action must be a neighbor of i, and thus the local game of the optimal action must contain the second best action.

Finally, thanks to the construction of p_t, the two-level sampling of I_t is equivalent to just sampling I_t from p_t. To see why this is true, let us calculate the distribution of I_t given the two-level sampling:

$$\begin{aligned}\mathbb{P}\left(I_t = i\right) &= \sum_{j=1}^{N} p_t(j)\mathbb{I}_{\{i\in N_j\}} q_t^j(i) \\ &= \sum_{j=1}^{N} p_t(j) q_t^j(i) = [Q_t p_t](i) = p_t(i)\,,\end{aligned}$$

since $p_t = Q_t p_t$. This property proves to be useful when analyzing the algorithm: one can think of p_t as both the probability distribution of playing in local games as well as the probability distribution of choosing the actions.

Now it is time to turn our attention to how the local algorithms work. As said earlier, the local algorithms maintain the vectors q_t^i. These vectors are generated based on unbiased estimates of the loss differences between i and the other actions in N_i. These estimates are calculated based on the formula

$$b_t^{i,j} = \mathbb{I}_{\{I_t=i\}} v_{i,j,i}^\top Y_t + \mathbb{I}_{\{k_t=i,I_t=j\}} v_{i,j,j}^\top Y_t / q_t^i(j)\,,$$

where k_t is the local game selected at time step t. Recall that the vectors $v_{\cdot,\cdot,\cdot}$ are the observer vectors defined in Section 6.2.1, while $Y_t = S_{I_t} e_{J_t}$ is the feedback vector. It is not hard to see that the conditional expectation (conditioned on the past observations and actions) of $b_t^{i,j}$ is the loss difference of actions i and j at time step t. With the help of these estimates, all the local algorithms update their q_t^i distributions based on exponential weighting, and then I_t is chosen by the selected algorithm $\mathcal{A}_{k_t}$. The pseudocodes for the local algorithms and for the meta-algorithm can be found in Algorithms 16 and 17.

Foster and Rakhlin [2011] show that the algorithm NeighborhoodWatch achieves $O(\sqrt{T})$ minimax regret on locally observable games against adversarial opponents. The main steps of their analysis are:

- Show that the local algorithms achieve low regret on the corresponding local games.
- Show that it is implied then that the *local internal regret* of NeighborhoodWatch is low.
- Exploit that the *internal regret*[6] is always smaller than the local internal regret and that the regret is always smaller than the internal regret.

With the algorithm NeighborhoodWatch and its analysis, Foster and Rakhlin showed that our conjecture (Conjecture 1) is true for non-degenerate finite partial-monitoring games.

[6]We do not include definitions for local internal regret and internal regret here, but refer the reader to Foster and Rakhlin [2011] for these definitions.

Algorithm 16 NeighborhoodWatch Local i (taken from Foster and Rakhlin [2011]

Initialize $w^i = e_{N_i}$
for $t = 1 : T$ **do**
  $q^i \leftarrow w^i / \|w^i\|_1$
  Receive k_t from meta-algorithm
  if $k_t = i$ **then**
    Choose action I_t with distribution q^i
  end if
  Receive observation Y
  for each $j \in N_i$ **do**
    $b^{i,j} \leftarrow \mathbb{I}_{\{I_t=i\}} v_{i,j,i}^\top Y + \mathbb{I}_{\{k_t=i, I_t=j\}} v_{i,j,j}^\top Y / q_t^i(j)$
    $w^i(j) \leftarrow w^i(j) \exp(-\eta b^{i,j})$
  end for
end for

Algorithm 17 NeighborhoodWatch Meta (taken from Foster and Rakhlin [2011])

for $t = 1 : T$ **do**
  Receive q^i from local algorithms
  Construct $Q = \begin{pmatrix} q^1 & q^2 & \cdots & q^N \end{pmatrix}$.
  Construct fixed-point $p \in \Delta_N$ by solving the equation $p = Qp$
  Choose local game k_t following distribution p
  Receive I_t from local algorithm $\mathcal{A}_{k_t}$
end for

Chapter 7

Better algorithms for finite stochastic games[1]

The algorithm Balaton described in Section 6.2.1 was designed solely for the purpose of proving the classification theorem. In particular, it shows that if a finite partial-monitoring game satisfies the local observability condition (Definition 10) then it is possible to achieve $\widetilde{O}(\sqrt{T})$ regret against a stochastic opponent. Unfortunately, apart from being important from a theoretical point of view, Balaton is not a very practical algorithm. The constant in the regret bound is very large and it needs to know the time horizon to achieve the root-T regret.[2] These drawbacks motivated us to design a new, better algorithm that has better practical performance. Our desire was to have an algorithm that does not need the time horizon as input and works better in practice. Furthermore, intuition suggested that easy games could have logarithmic individual regret bounds, just like bandit games. Thus, an extra "wish" was that the new algorithm achieve logarithmic individual regret. In this chapter we introduce two new algorithms, CBP-Vanilla and CBP[3], and prove that they have some very desirable properties.

7.1 An anytime algorithm with logarithmic individual regret: CBP-Vanilla

In this section we describe the algorithm that, for every locally observable (easy) game, achieves logarithmic individual expected regret, as well as optimal minimax regret (up to logarithmic factors).

In a nutshell, the algorithm works as follows. For every neighboring ac-

[1] Part of this chapter is based on the work by Bartók, Zolghadr, and Szepesvári [2012] to be published at ICML2012.

[2] One can always use the "doubling trick" to overcome this disadvantage, but that increases the constant factor further, and makes the algorithm even less usable.

[3] The letters CBP stand for "Confidence Bound Partial monitoring". The "-Vanilla" foreshadows that in subsequent sections this algorithm will be further improved, to have even more advantageous properties.

tion pair it maintains an unbiased estimate of the expected difference of their losses. It also keeps a confidence width for these estimates. If at time step t an estimate is "confident enough" to determine which action is better, the algorithm excludes some actions from the set of potentially optimal actions: for example, if the estimate for the pair i, j is confident and action i is estimated to have smaller loss, then we know that $(\ell_j - \ell_i)^\top p^* > 0$ and thus actions whose cells lie completely in the halfspace $\{p \ : \ (\ell_j - \ell_i)^\top p \leq 0\}$ can not be optimal (with high confidence). Doing this exclusion for every pair with a confident loss difference estimate, we arrive at a set of actions that are candidates for being optimal. Within this set, we enumerate the neighboring action pairs and collect their neighborhood action set. Then, we choose the action within this set that reduces the confidence widths the most.

At any time step t, the estimate of the loss difference of actions i and j is calculated as

$$\tilde{\delta}_{i,j}(t) = \sum_{k \in N^+_{i,j}} v^\top_{i,j,k} \frac{\sum_{s=1}^{t-1} \mathbb{I}_{\{I_s = k\}} Y_s}{\sum_{s=1}^{t-1} \mathbb{I}_{\{I_s = k\}}} \,,$$

similarly as for the algorithm Balaton. The confidence bound of the loss difference estimate is defined as

$$c_{i,j}(t) = \sum_{k \in N^+_{i,j}} \|v_{i,j,k}\|_\infty \sqrt{\frac{\alpha \log t}{\sum_{s=1}^{t-1} \mathbb{I}_{\{I_s = k\}}}}$$

with some preset parameter α. We call the estimate $\tilde{\delta}_{i,j}(t)$ confident if $|\tilde{\delta}_{i,j}(t)| \geq c_{i,j}(t)$.

By default, the set of candidate actions is not all actions but only the set of Pareto-optimal actions, denoted by $\mathcal{P}$. Then, at time step t, this set is narrowed by excluding confidently suboptimal actions. We also keep track of the current neighboring action pairs $\mathcal{N}(t)$. This is important, because some action pairs cease to be neighbors when we exclude a region from the probability simplex. Then, the set of possible actions $Q(t)$ is defined as the union of the neighboring action sets of pairs in $\mathcal{N}(t)$:

$$Q(t) = \bigcup_{\{i,j\} \in \mathcal{N}(t)} N^+_{i,j} \,.$$

Finally, the action is chosen to be the one that potentially reduces the confidence widths the most:

$$I_t = \mathrm{argmax}_{k \in Q(t)} \frac{W_k^2}{\sum_{s=1}^{t-1} \mathbb{I}_{\{I_s = k\}}} \,,$$

where $W_k = \max\{\|v_{i,j,k}\|_\infty \ : \ k \in N^+_{i,j}\}$ with fixed $v_{i,j,k}$ precomputed and used by the algorithm. Pseudocode for the algorithm is given in Algorithm 18.

Symbol	*Definition*
$N, M \in \mathbb{N}$	number of actions and outcomes
$\underline{N}$	$\{1, \dots, N\}$, set of actions
$\Delta_M \subset \mathbb{R}^M$	M-dim. simplex, set of opponent strategies
$p^* \in \Delta_M$	opponent strategy
$\mathbf{L} \in \mathbb{R}^{N \times M}$	loss matrix
$\mathbf{H} \in \Sigma^{N \times M}$	feedback matrix
$\ell_i \in \mathbb{R}^M$	$\ell_i = \mathbf{L}[i,:]$, loss vector underlying action i
$C_i \subseteq \Delta_M$	cell of action i
$\mathcal{P} \subseteq \underline{N}$	set of Pareto-optimal actions
$\mathcal{N} \subseteq \underline{N}^2$	set of unordered neighboring action-pairs
$N^+_{i,j} \subseteq \underline{N}$	neighborhood action set of $\{i,j\} \in \mathcal{N}$
$S_i \in \{0,1\}^{s_i \times M}$	signal matrix of action i
$\mathcal{L} \subseteq \mathcal{N}$	set of locally observable action pairs
$V_{i,j} \subseteq \underline{N}$	observer actions underlying $\{i,j\} \in \mathcal{N}$
$v_{i,j,k} \in \mathbb{R}^{s_k}$, $k \in V_{i,j}$	observer vectors
$W_i \in \mathbb{R}$	confidence width for action $i \in \underline{N}$

Table 7.1: List of basic symbols

It remains to specify the function GETPOLYTOPE. It gets the array *halfSpace* as input. The array *halfSpace* stores which neighboring action pairs have a confident estimate on the difference of their expected losses, along with the sign of the difference (if confident). Each of these confident pairs define an open halfspace, namely

$$\Delta_{\{i,j\}} = \left\{ p \in \Delta_M \,:\, halfSpace(i,j)(\ell_i - \ell_j)^\top p > 0 \right\} .$$

The function GETPOLYTOPE calculates the open polytope defined as the intersection of the above halfspaces. Then for all $i \in \mathcal{P}$ it checks if C_i intersects with the open polytope. If so, then i will be an element of $\mathcal{P}(t)$. Similarly, for every $\{i,j\} \in \mathcal{N}$, it checks if $C_i \cap C_j$ intersects with the open polytope and puts the pair in $\mathcal{N}(t)$ if it does.

Note that it is not enough to compute $\mathcal{P}(t)$ and then drop from $\mathcal{N}$ those pairs $\{k,l\}$ where one of k or l is excluded from $\mathcal{P}(t)$: it is possible that the boundary $C_k \cap C_l$ between the cells of two actions $k, l \in \mathcal{P}(t)$ is included in the rejected region.

For the convenience of the reader, we include a list of symbols used in this Chapter in Table 7.1.

7.1.1 Analysis of the algorithm

In this section we prove individual and minimax upper bounds on the expected regret of the algorithm.

Algorithm 18 CBP-Vanilla

Input: $\mathbf{L}$, $\mathbf{H}$, α
Calculate P, $\mathcal{N}$, $N_{i,j}^+$, $v_{i,j,k}$, W_k
for $t = 1$ **to** N **do**
  Choose $I_t = t$ {Initialization}
  Observe Y_t
  $n_{I_t} \leftarrow 1$ {# times action is chosen}
  $\nu_{I_t} \leftarrow Y_t$ {cumulative observations}
end for
for $t = N + 1$ **to** T **do**
  $\mathcal{P}(t) \leftarrow \mathcal{P}$ {Plausible actions}
  $\mathcal{N}(t) \leftarrow \mathcal{N}$ {Neighboring plausible actions}
  for each $\{i, j\} \in \mathcal{N}$ **do**
    $\tilde{\delta}_{i,j} \leftarrow \sum_{k \in V_{i,j}} v_{i,j,k}^\top \frac{\nu_k}{n_k}$ {Loss diff. estimate}
    $c_{i,j} \leftarrow \sum_{k \in V_{i,j}} \|v_{i,j,k}\|_\infty \sqrt{\frac{\alpha \log t}{n_k}}$ {Confidence}
    if $|\tilde{\delta}_{i,j}| \geq c_{i,j}$ **then**
      $halfSpace(i, j) \leftarrow \operatorname{sgn} \tilde{\delta}_{i,j}$
    else
      $halfSpace(i, j) \leftarrow 0$
    end if
  end for
  $[\mathcal{P}(t), \mathcal{N}(t)] \leftarrow \textsc{getPolytope}(\mathcal{P}, \mathcal{N}, halfSpace)$
  $Q \leftarrow \{k \ : \ \exists \{i, j\} \in \mathcal{N}(t) \text{ s.t. } k \in N_{i,j}^+\}$ {Admissible actions}
  Choose $I_t = \operatorname{argmax}_{i \in Q} \frac{W_i^2}{n_i}$
  Observe Y_t
  $\nu_{I_t} \leftarrow \nu_{I_t} + Y_t$
  $n_{I_t} \leftarrow n_{I_t} + 1$
end for

Theorem 12. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be an N by M partial-monitoring game. For a fixed opponent strategy $p^* \in \Delta_M$, let δ_i denote the difference between the expected loss of arm i and an optimal action. For any time horizon T, algorithm* CBP *with parameter $\alpha > 1$ has expected regret*

$$\mathbb{E}[R_T] \leq 2 \sum_{\{i,j\} \in \mathcal{N}} |N_{i,j}^+| \left(1 + \frac{1}{2\alpha - 2}\right) + \sum_{i=1}^{N} \delta_i + 4 \sum_{k : \delta_k > 0} W_k^2 \frac{d_k^2}{\delta_k} \alpha \log T \,.$$

Proof. We use the convention that, for any variable x used by the algorithm, $x(t)$ denotes the value of x at the end of time step t. For example, $n_i(t)$ is the number of times action i is chosen up to time step t.

For two actions i and j, let $\delta_{i,j}$ be the difference of their expected losses, that is, $\delta_{i,j} = (\ell_i - \ell_j)^\top p^*$.

Before getting into the proof, we need a lemma. The lemma shows that the estimate $\tilde{\delta}_{i,j}(t)$ is in the vicinity of $\delta_{i,j}$ with high probability.

Lemma 18. *For any $\{i,j\} \in \mathcal{N}$, $t \geq 1$,*

$$\mathbb{P}\left(|\tilde{\delta}_{i,j}(t) - \delta_{i,j}| \geq c_{i,j}(t)\right) \leq 2|N_{i,j}^+|t^{1-2\alpha}\,.$$

By Wald's identity, we can rewrite the expected regret as follows:

$$\begin{aligned}\mathbb{E}[R_T] &= \mathbb{E}\left[\sum_{t=1}^{T} L[I_t, J_T]\right] - \min_{i \in \underline{N}} \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{L}[i, J_t]\right] \\ &= \sum_{i=1}^{N} \mathbb{E}[n_i(T)]\delta_i\,.\end{aligned}$$

Hence, we need an upper bound on $\mathbb{E}[n_i(T)]$ for every suboptimal action.

Let action k be suboptimal. The number of times action k is chosen can be written as

$$n_k(T) = \sum_{t=1}^{T} \mathbb{I}_{\{I_t = k\}}\,.$$

At any time step t, action k can be chosen by the following reasons:

1. The algorithm is in the first **for** loop, that is, $t = k$.
2. Some confidence widths fail. The event of failure at time step t will be called $\mathcal{E}_t$.
3. Action k is in $Q(t)$ with $W_k^2/n_k(t-1) \geq W_l^2/n_l(t-1)$ for every $l \in Q(t)$.

Thus,

$$n_k(T) = 1 + \sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t, I_t = k\}} + \sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c, I_t = k\}}$$

implying

$$\mathbb{E}[n_k(T)] \leq 1 + \sum_{t=N+1}^{T} \mathbb{P}\left(\mathcal{E}_t\right) + \sum_{t=N+1}^{T} \mathbb{P}\left(\mathcal{E}_t^c, I_t = k\right)\,.$$

First, with the help of Lemma 18, we upper bound the probability that any confidence interval fails at time step t.

$$\begin{aligned}\mathbb{P}\left(\mathcal{E}_t\right) &= \mathbb{P}\left(\exists\{i,j\} \in \mathcal{N} \;:\; |\tilde{\delta}_{i,j}(t) - \delta_{i,j}| \geq c_{i,j}(t)\right) \\ &\leq 2 \sum_{\{i,j\} \in \mathcal{N}} |N_{i,j}^+|t^{1-2\alpha}\,.\end{aligned}$$

To continue the upper bounding of $\mathbb{E}[n_k(T)]$ we write

$$\begin{aligned}
\mathbb{E}[n_k(T)] &\le 1 + \sum_{t=N+1}^{T} \mathbb{P}\left(\mathcal{E}_t\right) \\
&\quad + \sum_{t=N+1}^{T} \mathbb{P}\left(\mathcal{E}_t^c, I_t = k\right) \\
&\le 1 + 2 \sum_{\{i,j\}\in\mathcal{N}} |V_{i,j}| \sum_{t=N+1}^{T} t^{1-2\alpha} \\
&\quad + \sum_{t=N+1}^{T} \mathbb{P}\left(\mathcal{E}_t^c, I_t = k\right) \\
&\le 1 + 2 \sum_{\{i,j\}\in\mathcal{N}} |V_{i,j}| \left(1 + \frac{1}{2\alpha - 2}\right) \\
&\quad + A_k + \sum_{t=N+1}^{T} \mathbb{P}\left(\mathcal{E}_t^c, I_t = k, n_k(t) > A_k\right)
\end{aligned}$$

for any positive value of A_k. The rest of the proof is devoted to finding an A_k value that gives an appropriate upper bound. In fact, we will find A_k such that the last term of the above bound is zero.

We observe that for any neighboring action pair $\{i,j\} \in \mathcal{N}(t)$, $\delta_{i,j} \le 2c_{i,j}(t)$. Since $I_t = k$, we have that for all $k' \in N_{i,j}^+$, $\|v_{i,j,k'}\|_\infty / \sqrt{n_{k'}(t-1)} \le W_k / \sqrt{n_k(t-1)}$, and thus $\delta_{i,j} \le 2|N_{i,j}^+| W_k \sqrt{\frac{\alpha \log t}{n_k(t-1)}}$.

To prepare for the next lemma, we need some new notations and a definition.

Definition 11. *Let us denote the dependence of the random sets $\mathcal{P}(t)$, $\mathcal{N}(t)$ on the outcomes ω from the underlying sample space Ω by $\mathcal{P}_\omega(t)$ and $\mathcal{N}_\omega(t)$. With this, we define the* set of plausible configurations *to be*

$$\Psi = \cup_{t\ge 1} \left\{ (\mathcal{P}_\omega(t), \mathcal{N}_\omega(t)) \,:\, \omega \in \mathcal{E}_t^c \right\} .$$

Call $\pi = (i_0, i_1, \ldots, i_r)$ $(r \ge 0)$ a path *in $\mathcal{N}' \subseteq \underline{N}^2$ if $\{i_s, i_{s+1}\} \in \mathcal{N}'$ for all $0 \le s \le r-1$ (when $r = 0$ there is no restriction on π). The path is said to start at i_0 and end at i_r. Denoting by i^* an optimal action under p^* (i.e., $\ell_{i^*}^\top p^* \le \ell_i^\top p^*$ holds for all actions i), the set of paths that connect i to i^* and lie in $\mathcal{N}'$ will be denoted by $B_i(\mathcal{N}')$.*

The next lemma shows that $B_i(\mathcal{N}')$ is non-empty whenever $\mathcal{N}'$ is such that for some $\mathcal{P}'$, $(\mathcal{P}', \mathcal{N}') \in \Psi$:

Lemma 19. *Take an action i and a plausible pair $(\mathcal{P}', \mathcal{N}') \in \Psi$ such that $i \in \mathcal{P}'$. Then there exists a path π that starts at i and ends at i^* that lies in $\mathcal{N}'$.*

For $i \in \mathcal{P}$ define

$$d_i = \max_{\substack{(\mathcal{P}',\mathcal{N}')\in\Psi \\ i\in\mathcal{P}'}} \min_{\substack{\pi\in B_i(\mathcal{N}') \\ \pi=(i_0,\ldots,i_r)}} \sum_{s=1}^{r} |N^+_{i_{s-1},i_s}| \,.$$

According to the previous lemma, for each Pareto-optimal action i, the quantity d_i is well-defined and finite. The definition is extended to degenerate actions by defining d_i to be $\max(d_l, d_k)$, where k, l are such that $i \in N^+_{k,l}$.

The following lemma is the key step in finding the right A_k value.

Lemma 20. *Take any action k. On the event $\mathcal{E}^c$, from $I_t = k$ it follows that*

$$n_k(t-1) \le 4W_k^2 \frac{d_k^2}{\delta_k^2}\,\alpha \log t \,.$$

Now choosing A_k as $A_k = 4W_k^2 \frac{d_k^2}{\delta_k^2}\,\alpha \log T$, we get

$$\begin{aligned}\mathbb{E}[R_T] \le 2 \sum_{\{i,j\}\in\mathcal{N}} |N^+_{i,j}| \left(1 + \frac{1}{2\alpha - 2}\right) + \sum_{i=1}^{N} \delta_i \\ + 4 \sum_{k:\delta_k>0} W_k^2 \frac{d_k^2}{\delta_k}\,\alpha \log T \,.\end{aligned}$$

□

The next corollary of Theorem 12 upper bounds the minimax regret of any locally observable game.

Corollary 1. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ an N by M finite partial-monitoring game that satisies the local observability condition. Then there exists a constant C such that for any $p \in \Delta_M$, the algorithm* CBP-Vanilla *against opponent p has expected regret*

$$\mathbb{E}[R_T] \le C\sqrt{T \log T} \,.$$

Proof. For an arbitrary $\gamma > 0$, the result of Theorem 12 can be rewritten as

$$\begin{aligned}\mathbb{E}[R_T] &\le 2 \sum_{\{i,j\}\in\mathcal{N}} |N^+_{i,j}| \left(1 + \frac{1}{2\alpha - 2}\right) + \sum_{i=1}^{N} \delta_i \\ &\quad \sum_{k:\delta_k<\gamma} \gamma n_k(T) + 4 \sum_{k:\delta_k>\gamma} W_k^2 \frac{d_k^2}{\delta_k}\,\alpha \log T \\ &\le 2 \sum_{\{i,j\}\in\mathcal{N}} |N^+_{i,j}| \left(1 + \frac{1}{2\alpha - 2}\right) + \sum_{i=1}^{N} \delta_i \\ &\quad \gamma T + 4 \max_{k:\delta_k>\gamma} \left(W_k^2 d_k^2\right) \frac{\alpha \log T}{\gamma} \,.\end{aligned}$$

Setting

$$\gamma = 2 \max_k (W_k d_k) \sqrt{\frac{\alpha \log T}{T}}$$

gives the statement of the corollary. □

7.2 Improving CBP-Vanilla: an adaptive algorithm

From the previous section we see that it is possible to achieve logarithmic individual regret for easy partial-monitoring games. The bound in Theorem 12 shows that, just like for bandit games, the constant term in the bound depends on the "gaps" between the expected loss of optimal actions and that of suboptimal ones. Intuitively, this gap depends on how far the opponent strategy p^* is away from the boundaries of cells of Pareto-optimal actions. The question arises if it is possible to achieve similar results for games that are only globally observable. Is there an algorithm that can be run on both locally and globally observable games, recovers the result of CBP-Vanilla for locally observable games, while achieving reasonable (maybe near-optimal?) individual and minimax regret for globally observable games? To this end, in this section we introduce the algorithm CBP, which is a refined version of CBP-Vanilla.

The algorithm CBP If a game does not satisfy the local observability condition, it means that there exists a neighboring action pair such that the expected difference of their losses can not be estimated by using observations from actions within the neighborhood action set. However, we still would like to use an estimate similar to that of CBP-Vanilla. Luckily, the global observability condition (see Definition 9) ensures that, with the help of some extra actions outside of the neighborhood action set, estimating the expected loss difference is possible. This motivates the following definition:

Definition 12 (Observer sets and observer vectors)**.** *The* observer set $V_{i,j} \subset \underline{N}$ *underlying a pair of neighboring actions* $\{i, j\} \in \mathcal{N}$ *is a set of actions such that*

$$\ell_i - \ell_j \in \oplus_{k \in V_{i,j}} \operatorname{Im} S_k^\top .$$

The observer vectors $(v_{i,j,k})_{k \in V_{i,j}}$ *underlying* $V_{i,j}$ *are defined to satisfy the equation* $\ell_i - \ell_j = \sum_{k \in V_{i,j}} S_k^\top v_{i,j,k}$*. In particular,* $v_{i,j,k} \in \mathbb{R}^{s_k}$*. In what follows, the choice of the observer sets and vectors is restricted so that* $V_{i,j} = V_{j,i}$ *and* $v_{i,j,k} = -v_{j,i,k}$*. Furthermore, the observer set* $V_{i,j}$ *is constrained to be a superset of* $N_{i,j}^+$ *and in particular when a pair* $\{i, j\}$ *is locally observable,* $V_{i,j} = N_{i,j}^+$ *must hold. Finally, for any action* $k \in \bigcup_{\{i,j\} \in \mathcal{N}} V_{i,j}$*, let* $W_k = \max_{i,j:k \in V_{i,j}} \|v_{i,j,k}\|_\infty$ *be the* confidence width *of action* k*.*

The reason for the particular choice of $V_{i,j} = N^+_{i,j}$ for locally observable pairs $\{i,j\}$ is that we plan to use $V_{i,j}$ (and the vectors $v_{i,j,.}$) in the case of not locally observable pairs, too. For not locally observable pairs, the whole action set $\underline{N}$ is always a valid observer set (thus, $V_{i,j}$ can be found). However, whenever possible, it is better to use a smaller set. The actual choice of $V_{i,j}$ (and $v_{i,j,k}$) is postponed until the effect of this choice on the regret becomes clear.

Now, one could run CBP-Vanilla replacing the neighborhood action sets with the observer sets. On locally observable games, it gives the same result as the original CBP-Vanilla, since the observer sets are defined to be the same as the neighborhood action sets for locally observable neighboring action pairs. However, if we run it on a not locally observable game, there is one more obstacle to overcome. Consider the case when the opponent strategy is in $C_i \cap C_j$ for $\{i,j\} \in \mathcal{N} \setminus \mathcal{L}$, that is, it is on the boundary between two non-locally observable neighboring actions. Unfortunately, our algorithm (CBP-Vanilla with observer sets) will suffer linear regret! The reason is that in this case both actions i and j are optimal, thus they never get eliminated, making the algorithm choose actions from $V_{i,j} \setminus N^+_{i,j}$ too often. Furthermore, even if the opponent strategy is not on the boundary the regret can be too high: say action i is optimal but δ_j is small, while $\{i,j\} \in \mathcal{N} \setminus \mathcal{L}$. Then a third action $k \in V_{i,j}$ with potentially large δ_k will be chosen proportional to $1/\delta_j^2$ times, causing high regret.

To combat the above phenomenon, we restrict the frequency with which an action can be used for "information seeking purposes". For this, we introduce the set of rarely chosen actions,

$$\mathcal{R}(t) = \{k \in \underline{N} \,:\, n_k(t) \le \eta_k f(t)\}\,,$$

where $\eta_k \in \mathbb{R}$, $f : \mathbb{N} \to \mathbb{R}$ are tuning parameters to be chosen later. Then, the set of actions available at time t is restricted to $\mathcal{P}(t) \cup N^+(t) \cup (\mathcal{V}(t) \cap \mathcal{R}(t))$, where $N^+(t) = \bigcup_{\{i,j\}\in\mathcal{N}(t)} N^+_{i,j}$ and $\mathcal{V}(t) = \bigcup_{\{i,j\}\in\mathcal{N}(t)} V_{i,j}$.

Pseudocode for the algorithm is given in Algorithm 19. The list of symbols used in the algorithm is shown in Table 7.2.

In the next section we prove regret bounds for the new algorithm CBP under various assumptions. The main result there is an individual regret bound for any globally observable game (see Theorem 13). This theorem yields the corollary that upper bounds the minimax regret of globally observable games (Corollary 2). Theorem 14 shows that CBP has an unexpected advantageous property: even for not locally observable games, if the opponent is "benign" in the sense that the set of possible strategies is isolated from boundaries between not locally observable action pairs, the algorithm achieves $\widetilde{O}(\sqrt{T})$ expected regret.

Algorithm 19 CBP

Input: $\mathbf{L}$, $\mathbf{H}$, α, $\eta_1, \dots, \eta_N$, $f = f(\cdot)$
Calculate $\mathcal{P}$, $\mathcal{N}$, $V_{i,j}$, $v_{i,j,k}$, W_k
for $t = 1$ **to** N **do**
  Choose $I_t = t$ and observe Y_t {Initialization}
  $n_{I_t} \leftarrow 1$ {# times the action is chosen}
  $\nu_{I_t} \leftarrow Y_t$ {Cumulative observations}
end for
for $t = N+1, N+2, \dots$ **do**
  $\mathcal{P}(t) \leftarrow \mathcal{P}$ {Plausible actions}
  $\mathcal{N}(t) \leftarrow \mathcal{N}$ {Neighboring plausible actions}
  for each $\{i,j\} \in \mathcal{N}$ **do**
    $\tilde{\delta}_{i,j} \leftarrow \sum_{k \in V_{i,j}} v_{i,j,k}^\top \frac{\nu_k}{n_k}$ {Loss diff. estimate}
    $c_{i,j} \leftarrow \sum_{k \in V_{i,j}} \|v_{i,j,k}\|_\infty \sqrt{\frac{\alpha \log t}{n_k}}$ {Confidence}
    if $|\tilde{\delta}_{i,j}| \geq c_{i,j}$ **then**
      $halfSpace(i,j) \leftarrow \operatorname{sgn} \tilde{\delta}_{i,j}$
    else
      $halfSpace(i,j) \leftarrow 0$
    end if
  end for
  $[\mathcal{P}(t), \mathcal{N}(t)] \leftarrow \textsc{getPolytope}(\mathcal{P}, \mathcal{N}, halfSpace)$
  $N^+(t) = \cup_{\{i,j\} \in \mathcal{N}(t)} N_{ij}^+$ {Plausible neighborhood actions}
  $\mathcal{V}(t) = \cup_{\{i,j\} \in \mathcal{N}(t)} V_{ij}$ {Plausible observer actions}
  $\mathcal{R}(t) = \{k \in \underline{N} : n_k(t) \leq \eta_k f(t)\}$ {Rarely sampled actions}
  $\mathcal{S}(t) = \mathcal{P}(t) \cup N^+(t) \cup (\mathcal{V}(t) \cap \mathcal{R}(t))$ {Admissible actions}
  Choose $I_t = \operatorname{argmax}_{i \in \mathcal{S}(t)} \frac{W_i^2}{n_i}$ and observe Y_t
  $\nu_{I_t} \leftarrow \nu_{I_t} + Y_t$
  $n_{I_t} \leftarrow n_{I_t} + 1$
end for

7.2.1 Analysis of the algorithm

In this section we provide individual and minimax upper bounds on the expected regret of CBP. The first theorem is an individual upper bound on the regret.

Theorem 13. *Let* $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ *be an* N *by* M *partial-monitoring game. For a fixed opponent strategy* $p^* \in \Delta_M$*, let* δ_i *denote the difference between the expected loss of action* i *and an optimal action. For any time horizon* T*, algorithm* CBP *with parameters* $\alpha > 1$*,* $\eta_k = W_k^{2/3}$*,* $f(t) = \alpha^{1/3} t^{2/3} \log^{1/3} t$ *has*

Symbol	*Definition*
$I_t \in \underline{N}$	action chosen at time t
$Y_t \in \{0,1\}^{s_{I_t}}$	observation at time t
$\tilde{\delta}_{i,j}(t) \in \mathbb{R}$	estimate of $(\ell_i - \ell_j)^\top p \qquad (\{i,j\} \in \mathcal{N})$
$c_{i,j}(t) \in \mathbb{R}$	confidence width for pair $\{i,j\} \qquad (\{i,j\} \in \mathcal{N})$
$\mathcal{P}(t) \subseteq \underline{N}$	plausible actions
$\mathcal{N}(t) \subseteq \underline{N}^2$	set of admissible neighbors
$N^+(t) \subseteq \underline{N}$	$\cup_{\{i,j\} \in \mathcal{N}(t)} N^+_{i,j}$; admissible neighborhood actions
$\mathcal{V}(t) \subseteq \underline{N}$	$\cup_{\{i,j\} \in \mathcal{N}(t)} V_{i,j}$; admissible information seeking actions
$\mathcal{R}(t) \subseteq \underline{N}$	rarely sampled actions
$\mathcal{S}(t)$	$\mathcal{P}(t) \cup N^+(t) \cup (\mathcal{V}(t) \cap \mathcal{R}(t))$; admissible actions

Table 7.2: List of symbols used in the algorithm

expected regret

$$\begin{aligned}\mathbb{E}[R_T] \le \sum_{\{i,j\} \in \mathcal{N}} 2|V_{i,j}| \left(1 + \frac{1}{2\alpha - 2}\right) + \sum_{k=1}^{N} \delta_k \\ + \sum_{\substack{k=1 \\ \delta_k > 0}}^{N} 4W_k^2 \frac{d_k^2}{\delta_k} \alpha \log T \\ + \sum_{k \in \mathcal{V} \setminus N^+} \delta_k \min\left(4W_k^2 \frac{d_{l(k)}^2}{\delta_{l(k)}^2} \alpha \log T,\ \alpha^{1/3} W_k^{2/3} T^{2/3} \log^{1/3} T\right) \\ + \sum_{k \in \mathcal{V} \setminus N^+} \delta_k \alpha^{1/3} W_k^{2/3} T^{2/3} \log^{1/3} T \\ + 2 d_k \alpha^{1/3} W^{2/3} T^{2/3} \log^{1/3} T\,,\end{aligned}$$

where $W = \max_{k \in \underline{N}} W_k$, $\mathcal{V} = \cup_{\{i,j\} \in \mathcal{N}} V_{i,j}$, $N^+ = \cup_{\{i,j\} \in \mathcal{N}} N^+_{i,j}$, *and* $d_1, \ldots, d_N$ *are game-dependent constants.*

Proof. As in the proof of the regret bound for CBP-Vanilla, again we use the convention that, for any variable x used by the algorithm, $x(t)$ denotes the value of x at the *end* of time step t. For example, $n_i(t)$ is the number of times action i is chosen up to and including time step t.

The proof is based on two lemmas. The first lemma shows that the estimate $\tilde{\delta}_{i,j}(t)$ is in the vicinity of $\delta_{i,j}$ with high probability.

Lemma 21. *For any* $\{i,j\} \in \mathcal{N}$, $t \ge 1$,

$$\mathbb{P}\left(|\tilde{\delta}_{i,j}(t) - \delta_{i,j}| \ge c_{i,j}(t)\right) \le 2|V_{i,j}| t^{1-2\alpha}\,.$$

If for some t, i, j, the event happens whose probability is upper-bounded in Lemma 21, we say that a confidence interval fails. Let $\mathcal{E}_t$ be the event

that some confidence interval fails in time step t. An immediate corollary of Lemma 21 is that the sum of the probabilities that some confidence interval fails is small:

$$\sum_{t=1}^{T} \mathbb{P}\left(\mathcal{E}_t\right) \leq \sum_{t=1}^{T} \sum_{\{i,j\}\in\mathcal{N}} 2|V_{i,j}|t^{-2\alpha} \leq \sum_{\{i,j\}\in\mathcal{N}} 2|V_{i,j}| \left(1 + \frac{1}{2\alpha - 2}\right). \tag{7.1}$$

Let $k(t) = \operatorname{argmax}_{i\in\mathcal{P}(t)\cup V(t)} W_i^2/n_i(t-1)$. When $k(t) \neq I_t$ this happens because $k(t) \notin N^+(t)$ and $k(t) \notin \mathcal{R}(t)$, *i.e.,* the action $k(t)$ is a "purely" information seeking action that has been sampled frequently. When this holds we say that the *"decaying exploration rule is in effect at time step t"*. The corresponding event is denoted by $\mathcal{D}_t = \{k(t) \neq I_t\}$. Using the notation of Definition 11 and the result of Lemma 19 we can recycle the definition of d_i from the proof of Theorem 12; we redefine these values using observer sets instead of neighborhood action sets:

$$d_i = \max_{\substack{(\mathcal{P}',\mathcal{N}')\in\Psi \\ i\in\mathcal{P}'}} \min_{\substack{\pi\in B_i(\mathcal{N}') \\ \pi=(i_0,\ldots,i_r)}} \sum_{s=1}^{r} |\mathcal{V}_{i_{s-1},i_s}|\,.$$

Now we can state the following lemma:

Lemma 22. *Fix any $t \geq 1$.*

1. *Take any action i. On the event $\mathcal{E}_t^c \cap \mathcal{D}_t$,*[4] *from $i \in \mathcal{P}(t) \cup N^+(t)$ it follows that*

$$\delta_i \leq 2d_i \sqrt{\frac{\alpha \log t}{f(t)}} \max_{k\in\underline{N}} \frac{W_k}{\sqrt{\eta_k}}\,.$$

2. *Take any action k. On the event $\mathcal{E}_t^c \cap \mathcal{D}_t^c$, from $I_t = k$ it follows that*

$$n_k(t-1) \leq \min_{j\in\mathcal{P}(t)\cup N^+(t)} 4W_k^2 \frac{d_j^2}{\delta_j^2}\,\alpha \log t\,.$$

We are now ready to start the proof. By Wald's identity, we can rewrite the expected regret as follows:

$$\begin{aligned}
\mathbb{E}[R_T] &= \mathbb{E}\left[\sum_{t=1}^{T} L[I_t, J_t]\right] - \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{L}[i^*, J_1]\right] = \sum_{k=1}^{N} \mathbb{E}[n_k(T)]\delta_i \\
&= \sum_{k=1}^{N} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_{\{I_t=k\}}\right] \delta_k \\
&= \sum_{k=1}^{N} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_{\{I_t=k,\mathcal{E}_t\}}\right] \delta_k + \sum_{k=1}^{N} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_{\{I_t=k,\mathcal{E}_t^c\}}\right] \delta_k\,.
\end{aligned}$$

[4]Here and in what follows all statements that start with "On event X" should be understood to hold almost surely on the event. However, to minimize clutter we will not add the qualifier "almost surely".

Now,

$$\sum_{k=1}^{N} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_{\{I_t=k,\mathcal{E}_t\}}\right] \delta_k \leq \sum_{k=1}^{N} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_{\{I_t=k,\mathcal{E}_t\}}\right] \qquad \text{(because } \delta_k \leq 1\text{)}$$
$$= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{k=1}^{N} \mathbb{I}_{\{I_t=k,\mathcal{E}_t\}}\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_{\{\mathcal{E}_t\}}\right] = \sum_{t=1}^{T} \mathbb{P}\left(\mathcal{E}_t\right) .$$

Hence,

$$\mathbb{E}[R_T] \leq \sum_{t=1}^{T} \mathbb{P}\left(\mathcal{E}_t\right) + \sum_{k=1}^{N} \mathbb{E}[\sum_{t=1}^{T} \mathbb{I}_{\{I_t=k,\mathcal{E}_t^c\}}]\delta_k \, .$$

Here, the first term can be bounded using (7.1). Let us thus consider the elements of the second sum:

$$\mathbb{E}[\sum_{t=1}^{T} \mathbb{I}_{\{I_t=k,\mathcal{E}_t^c\}}]\delta_k \leq \delta_k +$$
$$\mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c,\mathcal{D}_t^c,k\in\mathcal{P}(t)\cup N^+(t),I_t=k\}}]\,\delta_k \tag{7.2}$$
$$+\,\mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c,\mathcal{D}_t^c,k\notin\mathcal{P}(t)\cup N^+(t),I_t=k\}}]\,\delta_k \tag{7.3}$$
$$+\,\mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c,\mathcal{D}_t,k\in\mathcal{P}(t)\cup N^+(t),I_t=k\}}]\,\delta_k \tag{7.4}$$
$$+\,\mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c,\mathcal{D}_t,k\notin\mathcal{P}(t)\cup N^+(t),I_t=k\}}]\,\delta_k \, . \tag{7.5}$$

The first δ_k corresponds to the initialization phase of the algorithm when every action gets chosen once. The next paragraphs are devoted to upper bounding the above four expressions (7.2)-(7.5). Note that, if action k is optimal, that is, if $\delta_k = 0$ then all the terms are zero. Thus, we can assume from now on that $\delta_k > 0$.

Term (7.2): Consider the event $\mathcal{E}_t^c \cap D_t^c \cap \{k \in \mathcal{P}(t) \cup N^+(t)\}$. We use case 2 of Lemma 22 with the choice $i = k$. Thus, from $I_t = k$, we get that $i = k \in \mathcal{P}(t) \cup N^+(t)$ and so the conclusion of the lemma gives

$$n_k(t-1) \leq A_k(t) \stackrel{\triangle}{=} 4W_k^2 \frac{d_k^2}{\delta_k^2} \alpha \log t \, .$$

Therefore, we have

$$\begin{aligned}
&\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c, \mathcal{D}_t^c, k \in \mathcal{P}(t) \cup N^+(t), I_t = k\}} \\
&\quad \leq \sum_{t=N+1}^{T} \mathbb{I}_{\{I_t = k, n_k(t-1) \leq A_k(t)\}} \\
&\qquad + \sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c, \mathcal{D}_t^c, k \in \mathcal{P}(t) \cup N^+(t), I_t = k, n_k(t-1) > A_k(t)\}} \\
&\quad = \sum_{t=N+1}^{T} \mathbb{I}_{\{I_t = k, n_k(t-1) \leq A_k(t)\}} \\
&\quad \leq A_k(T) = 4W_k^2 \frac{d_k^2}{\delta_k^2} \alpha \log T
\end{aligned}$$

yielding

$$(7.2) \leq 4W_k^2 \frac{d_k^2}{\delta_k} \alpha \log T \,.$$

Term (7.3): Consider the event $\mathcal{E}_t^c \cap D_t^c \cap \{k \notin \mathcal{P}(t) \cup N^+(t)\}$. We use case 2 of Lemma 22. The lemma gives that

$$n_k(t-1) \leq \min_{j \in \mathcal{P}(t) \cup N^+(t)} 4W_k^2 \frac{d_j^2}{\delta_j^2} \alpha \log t \,.$$

We know that $k \in \mathcal{V}(t) = \cup_{\{i,j\} \in \mathcal{N}(t)} V_{i,j}$. Let Φ_t be the set of pairs $\{i, j\}$ in $\mathcal{N}(t) \subseteq \mathcal{N}$ such that $k \in V_{i,j}$. For any $\{i, j\} \in \Phi_t$, we also have that $i, j \in \mathcal{P}(t)$ and thus if $l'_{\{i,j\}} = \text{argmax}_{l \in \{i,j\}} \delta_l$ then

$$n_k(t-1) \leq 4W_k^2 \frac{d_{l'_{\{i,j\}}}^2}{\delta_{l'_{\{i,j\}}}^2} \alpha \log t \,.$$

Therefore, if we define $l(k)$ as the action with

$$\delta_{l(k)} = \min \left\{ \delta_{l'_{\{i,j\}}} \,:\, \{i, j\} \in \mathcal{N},\, k \in V_{i,j} \right\}$$

then it follows that

$$n_k(t-1) \leq 4W_k^2 \frac{d_{l(k)}^2}{\delta_{l(k)}^2} \alpha \log t \,.$$

Note that $\delta_{l(k)}$ can be zero and thus we use the convention $c/0 = \infty$. Also, since k is not in $\mathcal{P}(t) \cup N^+(t)$, we have that $n_k(t-1) \leq \eta_k f(t)$. Define $A_k(t)$ as

$$A_k(t) = \min \left(4W_k^2 \frac{d_{l(k)}^2}{\delta_{l(k)}^2} \alpha \log t, \eta_k f(t) \right) .$$

Then, with the same argument as in the previous case (and recalling that $f(t)$ is increasing), we get

$$(7.3) \leq \delta_k \min\left(4W_k^2 \frac{d_{l(k)}^2}{\delta_{l(k)}^2}\alpha \log T, \eta_k f(T)\right) .$$

We remark that without the concept of "rarely sampled actions", the above term would scale with $1/\delta_{l(k)}^2$, causing high regret. This is why the "vanilla version" of the algorithm fails on hard games.

Term (7.4): Consider the event $\mathcal{E}_t^c \cap D_t \cap \{k \in \mathcal{P}(t) \cup N^+(t)\}$. From case 1 of Lemma 22 we have that $\delta_k \leq 2d_k\sqrt{\frac{\alpha \log t}{f(t)}} \max_{j \in \underline{N}} \frac{W_j}{\sqrt{\eta_j}}$. Thus,

$$(7.4) \leq d_k T \sqrt{\frac{\alpha \log T}{f(T)}} \max_{l \in \underline{N}} \frac{W_l}{\sqrt{\eta_l}} .$$

Term (7.5): Consider the event $\mathcal{E}_t^c \cap D_t \cap \{k \notin \mathcal{P}(t) \cup N^+(t)\}$. Since $k \notin \mathcal{P}(t) \cup N^+(t)$ we know that $k \in \mathcal{V}(t) \cap \mathcal{R}(t) \subseteq \mathcal{R}(t)$ and hence $n_k(t-1) \leq \eta_k f(t)$. With the same argument as in the cases (7.2) and (7.3) we get that

$$(7.5) \leq \delta_k \eta_k f(T) .$$

To conclude the proof of Theorem 13, we set $\eta_k = W_k^{2/3}$, $f(t) = \alpha^{1/3} t^{2/3} \log^{1/3} t$ and, with the notation $W = \max_{k \in \underline{N}} W_k$, $\mathcal{V} = \cup_{\{i,j\} \in \mathcal{N}} V_{i,j}$, $N^+ = \cup_{\{i,j\} \in \mathcal{N}} N_{i,j}^+$, we write

$$\begin{aligned}
\mathbb{E}[R_T] \leq & \sum_{\{i,j\} \in \mathcal{N}} 2|V_{i,j}| \left(1 + \frac{1}{2\alpha - 2}\right) + \sum_{k=1}^{N} \delta_k \\
& + \sum_{\substack{k=1 \\ \delta_k > 0}}^{N} 4W_k^2 \frac{d_k^2}{\delta_k} \alpha \log T \\
& + \sum_{k \in \mathcal{V} \setminus N^+} \delta_k \min\left(4W_k^2 \frac{d_{l(k)}^2}{\delta_{l(k)}^2} \alpha \log T,\ \alpha^{1/3} W_k^{2/3} T^{2/3} \log^{1/3} T\right) \\
& + \sum_{k \in \mathcal{V} \setminus N^+} \delta_k \alpha^{1/3} W_k^{2/3} T^{2/3} \log^{1/3} T \\
& \qquad + 2d_k \alpha^{1/3} W^{2/3} T^{2/3} \log^{1/3} T .
\end{aligned}$$

□

The following corollary is an upper bound on the minimax regret of any globally observable game.

Corollary 2. *Let* $\mathbf{G}$ *be a globally observable game. Then there exists a constant* c *such that the expected regret can be upper bounded independently of the choice of* p^* *as*

$$\mathbb{E}[R_T] \leq c T^{2/3} \log^{1/3} T .$$

The following theorem is an upper bound on the minimax regret of any globally observable game against "benign" opponents. To state the theorem, we need a new definition. Let A be some subset of actions in $\mathbf{G}$. We call A a *point-local game* in $\mathbf{G}$ if $\bigcap_{i\in A} \mathcal{C}_i \neq \emptyset$.

Theorem 14. *Let* $\mathbf{G}$ *be a globally observable game. Let* $\Delta' \subseteq \Delta_M$ *be some subset of the probability simplex such that its topological closure* $\overline{\Delta'}$ *has* $\overline{\Delta'} \cap \mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ *for every* $\{i,j\} \in \mathcal{N} \setminus \mathcal{L}$. *Then there exists a constant* c *such that for every* $p^* \in \Delta'$, *algorithm* CBP *with parameters* $\alpha > 1$, $\eta_k = W_k^{2/3}$, $f(t) = \alpha^{1/3} t^{2/3} \log^{1/3} t$ *achieves*

$$\mathbb{E}[R_T] \leq c d_{pmax} \sqrt{bT \log T}\,,$$

where b *is the size of the largest point-local game, and* d_{pmax} *is a game-dependent constant.*

Proof. To prove this theorem, we use a scheme similar to the proof of Theorem 13. Repeating that proof, we arrive at the same expression

$$\begin{aligned}
\mathbb{E}[\sum_{t=1}^{T} \mathbb{I}_{\{I_t=k,\mathcal{E}_t^c\}}]\delta_k \leq \delta_k + \\
\mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c,\mathcal{D}_t^c,k\in\mathcal{P}(t)\cup N^+(t),I_t=k\}}]\,\delta_k & \quad (7.2)\\
+\,\mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c,\mathcal{D}_t^c,k\notin\mathcal{P}(t)\cup N^+(t),I_t=k\}}]\,\delta_k & \quad (7.3)\\
+\,\mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c,\mathcal{D}_t,k\in\mathcal{P}(t)\cup N^+(t),I_t=k\}}]\,\delta_k & \quad (7.4)\\
+\,\mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c,\mathcal{D}_t,k\notin\mathcal{P}(t)\cup N^+(t),I_t=k\}}]\,\delta_k\,, & \quad (7.5)
\end{aligned}$$

where $\mathcal{E}_t^c$ and $\mathcal{D}_t$ denote the events that no confidence intervals fail, and the decaying exploration rule is in effect at time step t, respectively.

From the condition of Δ' we have that there exists a positive constant ρ_1 such that for every neighboring action pair $\{i,j\} \in \mathcal{N} \setminus \mathcal{L}$, $\max(\delta_i, \delta_j) \geq \rho_1$. We know from Lemma 22 that if $\mathcal{D}_t$ happens then for any pair $\{i,j\} \in \mathcal{N} \setminus \mathcal{L}$ it holds that $\max(\delta_i, \delta_j) \leq 4N\sqrt{\frac{\alpha \log t}{f(t)}} \max(W_{k'}/\sqrt{\eta_{k'}}) \triangleq g(t)$. It follows that if $t > g^{-1}(\rho_1)$ then the decaying exploration rule can not be in effect. Therefore, terms (7.4) and (7.5) can be upper bounded by $g^{-1}(\rho_1)$.

With the value ρ_1 defined in the previous paragraph, we have that for any action $k \in \mathcal{V} \setminus N^+$, $l(k) \geq \rho_1$ holds. Therefore, term (7.3) can be upper bounded by

$$(7.3) \leq 4W^2 \frac{4N^2}{\rho_1^2} \alpha \log T\,,$$

using that d_k, defined in the proof of Theorem 13, is at most $2N$. It remains to carefully upper bound term (7.2). For that, we first need a definition and a lemma. Let $A_\rho = \{i \in \underline{N} \ : \ \delta_i \leq \rho\}$.

Lemma 23. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be a finite partial-monitoring game and $p \in \Delta_M$ an opponent strategy. There exists a $\rho_2 > 0$ such that A_{ρ_2} is a point-local game in $\mathbf{G}$.*

To upper bound term (7.2), with ρ_2 introduced in the above lemma and $\gamma > 0$ specified later, we write

$$\begin{aligned}
(7.2) &= \mathbb{E}[\sum_{t=N+1}^{T} \mathbb{I}_{\{\mathcal{E}_t^c, \mathcal{D}_t^c, k \in \mathcal{P}(t) \cup N^+(t), I_t = k\}}] \, \delta_k \\
&\leq \mathbb{I}_{\{\delta_k < \gamma\}} n_k(T) \delta_k + \mathbb{I}_{\{k \in A_{\rho_2}, \delta_k \geq \gamma\}} 4W_k^2 \frac{d_k^2}{\delta_k} \alpha \log T + \mathbb{I}_{\{k \notin A_{\rho_2}\}} 4W^2 \frac{8N^2}{\rho_2} \alpha \log T \\
&\leq \mathbb{I}_{\{\delta_k < \gamma\}} n_k(T) \gamma + |A_{\rho_2}| 4W^2 \frac{d_{pmax}^2}{\gamma} \alpha \log T + 4NW^2 \frac{8N^2}{\rho_2} \alpha \log T \,,
\end{aligned}$$

where d_{pmax} is defined as the maximum d_k value within point-local games.

Let b be the number of actions in the largest point-local game. Putting everything together we have

$$\begin{aligned}
\mathbb{E}[R_T] \leq \sum_{\{i,j\} \in \mathcal{N}} 2|V_{i,j}| \left(1 + \frac{1}{2\alpha - 2}\right) + g^{-1}(\rho_1) + \sum_{k=1}^{N} \delta_k \\
+ 16W^2 \frac{N^3}{\rho_1^2} \alpha \log T + 32W^2 \frac{N^3}{\rho_2} \alpha \log T \\
+ \gamma T + 4bW^2 \frac{d_{pmax}^2}{\gamma} \alpha \log T \,.
\end{aligned}$$

Now we choose γ to be

$$\gamma = 2W d_{pmax} \sqrt{\frac{b\alpha \log T}{T}}$$

and we get

$$\mathbb{E}[R_T] \leq c_1 + c_2 \log T + 4W d_{pmax} \sqrt{b\alpha T \log T} \,.$$

□

7.2.2 Example

In this section we demonstrate the results of the previous section through the example of Dynamic Pricing (see Example 6). Recall that in this game, a vendor (learner) tries to sell his product to a buyer (opponent). The buyer secretly chooses a maximum price (outcome) while the seller tries to sell it at

some price (action). If the outcome is lower than the action then no transaction happens and the seller suffers some constant loss. Otherwise the buyer buys the product and the seller's loss is the difference between the seller's price and the buyer's price. The feedback for the seller is, however, only the binary observation if the transaction happened (y for yes and n for no). The finite version of the game can be described with the following matrices:

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 2 & \cdots & N-1 \\ c & 0 & 1 & \cdots & N-2 \\ c & c & 0 & \cdots & N-3 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ c & \cdots & \cdots & c & 0 \end{pmatrix}; \qquad \mathbf{H} = \begin{pmatrix} y & y & \cdots & y \\ n & y & \cdots & y \\ \vdots & \ddots & \ddots & \vdots \\ n & \cdots & n & y \end{pmatrix}.$$

This game was proven to be hard in Chapter 6. That is, its minimax regret is $\Theta(T^{2/3})$.

As explained in the previous chapter, simple linear algebra gives that the locally observable action pairs are the "consecutive" actions ($\mathcal{L} = \{\{i, i+1\} : i \in \underline{N-1}\}$), while quite surprisingly, all action pairs are neighbors. In fact, there is a single point on the probability simplex that is common to all of the cells, namely

$$p = \begin{pmatrix} \frac{1}{c+1} & \frac{c}{(c+1)^2} & \cdots & \frac{c^{i-1}}{(c+1)^i} & \cdots & \frac{c^{N-2}}{(c+1)^{N-1}} & \frac{c^{N-1}}{(c+1)^{N-1}} \end{pmatrix}^\top.$$

It also follows that the game of dynamic pricing is a point-local game.

Now, we introduce a restriction on the space of opponent strategies such that the condition of Theorem 14 is satisfied. We need to prevent non-consecutive actions from being simultaneously optimal. A somewhat stronger condition is that out of three actions $i < j < k$, the loss of j should not be more than that of *both* i and k. We can prevent this from happening by preventing it for every triple $i-1, i, i+1$. Hence, a "bad" opponent strategy would satisfy

$$\ell_{i-1}^\top p \le \ell_i^\top p \qquad \text{and} \qquad \ell_{i+1}^\top p \le \ell_i^\top p\,.$$

After rearranging, the above two inequalities yield the constraints

$$p_i \le \frac{c}{c+1} p_{i-1}$$

for every $i = 2, \ldots, N-1$. Note that there is no constraint on p_N. If we want to avoid by a margin these inequalities to be satisfied, we arrive at the constraints

$$p_i \ge \frac{c}{c+1} p_{i-1} + \rho$$

for some $\rho > 0$, for every $i = 2, \ldots, N-1$.

In conclusion, we define the restricted opponent set to

$$\Delta' = \left\{ p \in \Delta_M \ : \ \forall i = 2, \ldots, N-2, \ p_i \geq \frac{c}{c+1} p_{i-1} + \rho \right\} .$$

The intuitive interpretation of this constraint is that the probability of the higher maximum price of the costumer should not decrease too fast. This constraint does not allow to have zero probabilities, and thus it is too restrictive.

Another way to construct a subset of Δ_M that is isolated from "dangerous" boundaries is to include only "hilly" distributions. We call a distribution $p \in \Delta_M$ hilly if it has a peak point $i^* \in \underline{N}$, and there exist $\xi_1, \ldots, \xi_{i^*-1} < 1$ and $\xi_{i^*+1}, \ldots, \xi_N < 1$ such that

$$\begin{aligned} p_{i-1} &\leq \xi_{i-1} p_i && \text{for } 2 \leq i \leq i^*, \text{ and} \\ p_{i+1} &\leq \xi_{i+1} p_i && \text{for } i^* \leq i \leq N-1. \end{aligned}$$

We now show that with the right choice of ξ_i, under a hilly distribution with peak i^*, only action i^* and maybe action $i^* - 1$ can be optimal.

1. If $i \leq i^*$ then

$$\begin{aligned} (\ell_i - \ell_{i-1})^\top p &= c p_{i-1} - (p_i + \cdots + p_N) \\ &\leq c \xi_{i-1} p_i - p_i - (p_{i+1} + \cdots + p_N) , \end{aligned}$$

   thus, if $\xi_{i-1} \leq 1/c$ then the expected loss of action i is less than or equal to that of action $i-1$.

2. If $i \geq i^*$ then

$$\begin{aligned} (\ell_{i+1} - \ell_i)^\top p &= c p_i - (p_{i+1} + \cdots + p_N) \\ &\geq p_i \left\{ c - (\xi_{i+1} + \xi_{i+1}\xi_{i+2} + \cdots + \prod_{j=i+1}^{N} \xi_j) \right\} . \end{aligned}$$

   Now if we let $\xi_{i^*+1} = \cdots = \xi_N = \xi$ then we get

$$\begin{aligned} (\ell_{i+1} - \ell_i)^\top p &\geq p_i \left(c - \xi \frac{1 - \xi^{N-1}}{1 - \xi} \right) \\ &\geq p_i \left(c - \frac{\xi}{1 - \xi} \right) , \end{aligned}$$

   and thus if we choose $\xi \leq \frac{c}{c+1}$ then the expected loss of action i is less than or equal to that of action $i+1$.

   So far in all the calculations we allowed equalities. If we want to achieve that only action i^* and possibly action $i^* - 1$ are optimal, we use

$$\xi_i \begin{cases} < 1/c, & \text{if } 2 \leq i \leq i^* - 2; \\ = 1/c, & \text{if } i = i^* - 1; \\ < c/(c+1), & \text{if } i^* + 1 \leq i \leq N. \end{cases}$$

If an opponent strategy is hilly with ξ_i satisfying all the above criteria, we call that strategy *sufficiently hilly*. Now we are ready to state the corollary of Theorem 14:

Corollary 3. *Consider the dynamic pricing game with N actions and M outcomes. If we restrict the set of opponent strategies Δ' to the set of all sufficiently hilly distributions then the minimax regret of the game is upper bounded by*

$$\mathbb{E}[R_T] \leq C\sqrt{T}$$

for some constant $C > 0$

Remark 2. *Note that the number of actions and outcomes $N = M$ does not appear in the bound because the size of the largest point local game with the restricted strategy set is always 2, irrespectively of the number of actions.*

7.2.3 Experiments

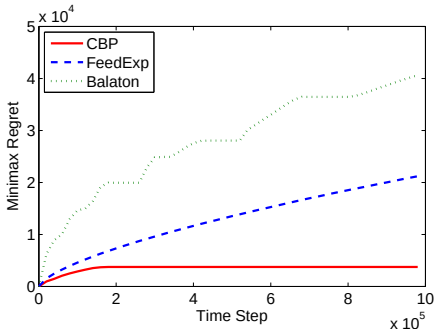
We demonstrate the results of the previous sections using instances of Dynamic Pricing, as well as a locally observable game. We compare the results of CBP to two other algorithms: Balaton (see Chapter 6) which is the first algorithm that achieves $\widetilde{O}(\sqrt{T})$ minimax regret for all locally observable finite stochastic partial-monitoring games; and FeedExp3 [Piccolboni and Schindelhauer, 2001], which achieves $O(T^{2/3})$ minimax regret on all non-hopeless finite partial-monitoring games, even against adversarial opponents.

A locally observable game

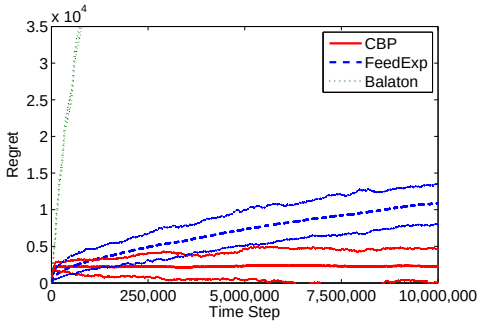
The game we use to compare CBP and Balaton has 3 actions and 3 outcomes. The game is described with the loss and feedback matrices:

$$\mathbf{L} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}; \qquad \mathbf{H} = \begin{pmatrix} a & b & b \\ b & a & b \\ b & b & a \end{pmatrix}.$$

We ran the algorithms 10 times for 15 different stochastic strategies. We averaged the results for each strategy and then took pointwise maximum over the 15 strategies. Figure 7.1(a) shows the empirical minimax regret calculated the way described above. In addition, Figure 7.1(b) shows the regret of the algorithms against one of the opponents, averaged over 100 runs. On the same figure, we also plotted the 90 percent empirical confidence intervals. The results indicate that CBP outperforms both FeedExp3 and Balaton. We also observe that, although the asymptotic performace of Balaton is proven to be better than that of FeedExp3, a larger constant factor makes Balaton lose against FeedExp3 even at time step ten million.

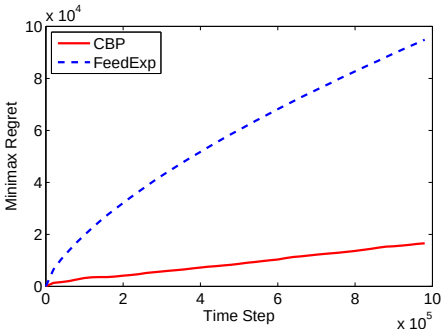


(a) Pointwise maximum over 15 settings.

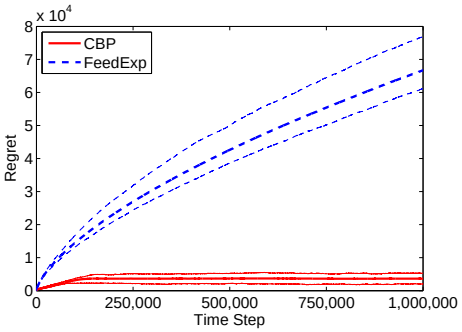


(b) Regret against one opponent strategy.

Figure 7.1: Comparing CBP with BALATON and FeedExp3 on the easy game



(a) Pointwise maximum over 15 settings.



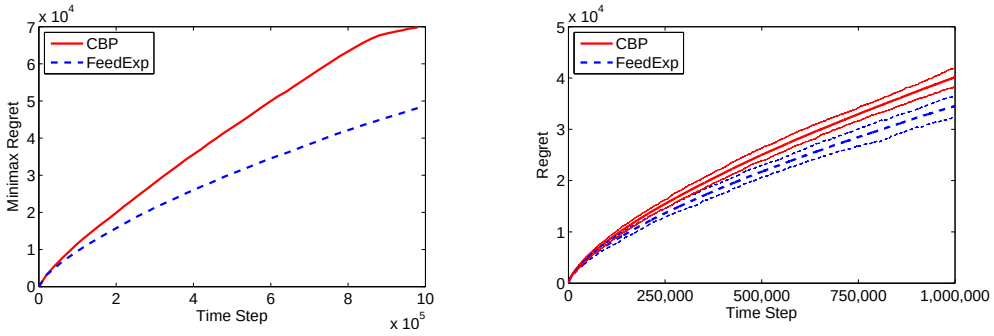
(b) Regret against one opponent strategy.

Figure 7.2: Comparing CBP and FeedExp3 on "benign" setting of the Dynamic Pricing game.

Dynamic Pricing

We compare CBP with FeedExp3 on Dynamic Pricing with $N = M = 5$ and $c = 2$. Since BALATON is undefined on not locally observable games, we can not include it in the comparison. To demonstrate the adaptiveness of CBP, we use two sets of opponent strategies. The "benign" setting is a set of opponents that are far away from "dangerous" regions, that is, from boundaries between cells of non-locally observable neighboring action pairs. The "harsh" settings include opponent strategies that are close or on the boundary between two such actions. For each setting we maximize over 15 strategies and average over 10 runs. We also compare the individual regret of the two algorithms against one benign and one harsh strategy. We averaged over 100 runs and plotted the 90 percent confidence intervals.

The results are shown in Figures 7.2 and 7.3. The figures clearly indicate that CBP has a significant advantage over FeedExp3 for the benign settings. On the other hand, for the harsh settings FeedExp3 slightly outperforms CBP, which we think is a reasonable price to pay for the benefit of adaptivity.



(a) Pointwise maximum over 15 settings. (b) Regret against one opponent strategy.

Figure 7.3: Comparing CBP and FeedExp3 on "harsh" setting of the Dynamic Pricing game.

Chapter 8

A case study: Online probe complexity[1]

In this chapter we introduce a new online learning game called *online probing* in which at each round, a player has to predict a label given some features. Before the player makes the prediction, he decides which features to observe, where every feature has an associated cost. After his prediction, the learner has the option to observe the true label, however, there is a cost assigned to observing the label as well. This way, the loss of the learner in each round has two components: the prediction error, and the cost of the features and the label.

We introduce two variations of the above game; in the first variant, observing the label is free, while in the second variant, requesting the true label has non-zero cost. We provide regret upper and lower bounds for both games and, by comparing the results for the two games, we show that a positive cost for asking the label significantly changes the complexity of the game.

8.1 The setting

In this section we define the online probing game. A game instance is defined by the cost of the features $c \in \mathbb{R}^d_+$, the cost of the label $c^L \geq 0$, and the prediction loss function $\hat{\ell}(\cdot, \cdot)$. We make the assumption that $\|c\|_1 \leq 1$. In every round, a feature vector $x_t \in [0,1]^d$ and a label $y_t \in [0,1]$ is chosen by an adversary. Initially, both of these are kept secret. Not knowing the feature vector, or the label, the learner decides which (if any) of the d features of the current example he wants to receive. We denote this decision by the binary vector $s_t \in \{0,1\}^d$: $s_{t,i} = 1$ means that the learner wants to see the value of feature i at time t. The learner also decides if he wants to see the label: we use $s^L_t \in \{0,1\}$ to denote this decision. Similarly to the previous case, $s^L_t = 1$ means that the learner wants to see the label at time t. Once the learner chooses s_t and s^L_t, the learner receives the values of the requested features,

[1] Joint work with Navid Zolghadr, Russ Greiner, and Csaba Szepesvári.

and makes a prediction $\hat{y}_t$. If $s_t^L = 1$ then the learner receives the true label y_t after he has made his prediction. The loss suffered by the learner is

$$\ell_t(I_t, \hat{y}_t, y_t) = \hat{\ell}(\hat{y}_t, y_t) + c^\top s_t + c^L s^L \,,$$

where we defined $I_t = (s_t, s_t^L)$. Whenever it is clear from the context, we will suppress the arguments of the loss function and simply write ℓ_t for the value of the loss suffered in round t. It is immediate from the problem setup that in rounds when the learner does not request the label, his loss is not revealed to him.

The goal of the learner is to minimize his cumulative loss $\sum_{t=1}^T \ell_t$. The performance of the learner is measured by the (cumulative) *regret*, defined as the excess cumulative loss of the learner compared to the best *linear predictor*. A linear predictor is defined by the pair (s, w) where $s \in \{0,1\}^d$ and $w \in [0,1]^d$. Its prediction is $\hat{y}_t = w^\top (s \odot x_t)$, where $\odot$ denotes the componentwise product. Thus, the regret of the learner is

$$R_T = \sum_{t=1}^T \ell_t(I_t, \hat{y}_t, y_t) - \inf_{(s,w)} \left\{ T c^\top s + \sum_{t=1}^T \hat{\ell}\left(w^\top (s \odot x_t), y_t\right) \right\} .$$

As we will see in the next sections, the "hardness" of a game dramatically depends on the cost of the label. We will show that, if the prediction loss function $\hat{\ell}$ is Lipschitz then, in the case when $c^L = 0$, *i.e.,* the label is free, the regret scales with the time horizon as $O(\sqrt{T})$ whereas, with positive label cost, the regret scales as $\Theta(T^{2/3})$. In the next sections, we present algorithms for these two versions of the game, as well as a lower bound for the latter one.

8.2 Free-label game

In this version of the game, we assume that $c^L = 0$. Hence, we can assume without loss of generality that the learner receives the true label at the end of each round. For brevity, we omit s_t^L from the action of the player.

We solve the game with the use of *experts*. An expert is defined by the pair (s, w). In each round, the expert (s, w) requests the features for which $s_i = 1$ and makes the prediction $\hat{y}_t = w^\top (s \odot x_t)$. In every round, the learner (randomly) chooses one of the experts.

The key property of this game is that if the learner chooses (s, w), he can also calculate the loss of other actions, namely, all actions with the same s. (Actually, losses of actions that choose a subset of the features the current action chooses can be calculated, but we will not use this fact.) Thus, the game is a hybrid of bandit and full-information game. As already mentioned in Chapter 2, Mannor and Shamir [2011] address this problem and introduce the algorithm ELP (for "Exponentially-weighted algorithm with Linear Programming"). Their result states that the expected regret of ELP can be upper

bounded by

$$\mathbb{E}[R_T] \leq C\sqrt{T\chi \log N}\,,$$

where N is the number of experts; and χ is the minimum number of cliques needed to cover all vertices of the graph of actions, where two action is connected with an edge if the loss for one can be recovered by choosing the other.

In our case, a clique is a set of experts with the same s value. The number of cliques is hence 2^d. However, Mannor and Shamir [2011] address the problem only in the case of finitely many actions, whereas in our case the number of experts is infinite. To overcome this problem, we use discretization. We discretize the space of w the usual way: given a discretization parameter $\alpha \in \mathbb{N}$, the set of experts is defined as

$$\mathcal{D}_\alpha = \left\{(s,w) \mid s \in \{0,1\}^d, \forall i \in \{1,\dots,d\}\, \exists \beta \in \{0,\dots,\alpha-1\} : w_i = \frac{\beta}{\alpha-1}\right\}.$$

It follows that the number of experts is $N = (2\alpha)^d$.

The following lemma upper bounds the approximation error caused by the discretization:

Lemma 24. *Given any expert (s,w) and label y, the* approximation error*, defined as $\min_{(s,w')\in\mathcal{D}_\alpha} |\hat{\ell}(w^\top(s\odot x), y) - \hat{\ell}(w'^\top(s\odot x), y)|$ is upper bounded by $\frac{L\sqrt{d}}{\alpha-1}$, where L is the Lipschitz constant of $\hat{\ell}$.*

Now we are ready to state the main theorem of this section.

Theorem 15. *There exists a constant C such that given an online probing game with $c^L = 0$, the "Exponentially-weighted algorithm with Linear Programming" [Mannor and Shamir, 2011] run on the set of experts $\mathcal{D}_\alpha$ has expected regret*

$$\mathbb{E}[R_T] \leq C\sqrt{T2^d \log(TL)}\,.$$

Proof. First we observe that the number of actions in the discretized version of the game is $(2\alpha)^d$, while χ (the number of cliques needed to cover all actions) is 2^d. The regret has two additive components: the regret of ELP compared to the best expert from $\mathcal{D}_\alpha$, and T times the approximation error:

$$\mathbb{E}[R_T] \leq C_1\sqrt{T2^d d \log(2\alpha)} + T\frac{L\sqrt{d}}{\alpha-1}\,.$$

Setting α to LT we get

$$\mathbb{E}[R_T] \leq C\sqrt{T2^d \ln(TL)}\,,$$

as stated in the theorem. □

Algorithm 20 Revealing action algorithm for non-free-label online probing

Parameters: Integer number $\alpha \geq 1$ and Real numbers $0 \leq \eta, \gamma \leq 1$
Initialization: Generate $\mathcal{D}_\alpha$, $\forall w \in \mathcal{D}_\alpha, u_0(w) \leftarrow 1$
for $t = 1$ **to** T **do**
  $U_{t-1} \leftarrow \sum_{w \in \mathcal{D}_\alpha} u_{t-1}(w)$
  Draw a Bernoulli random variable Z_t such that $\mathbb{P}(Z_t = 1) = \gamma$
  Draw w from distribution $p_t(w) = \frac{u_{t-1}(w)}{U_{t-1}}$
  if $Z_t = 0$ **then**
    Choose action $s_t = s(w_t), s_t^L = 0, w_t = w$
  else
    Choose action $s_t = \mathbf{1}, s_t^L = 1, w_t = w$
    Receive label y_t
  end if
  for each $w \in \mathcal{D}_\alpha$ **do**
    $\tilde{\ell}_t(w) \leftarrow \mathbb{I}_{\{Z_t=1\}} \frac{\ell(w, s(w), y_t)}{\gamma}$
    $u_t(w) \leftarrow u_{t-1}(w) \exp(-\eta \tilde{\ell}_t(w))$
  end for
end for

8.3 Non-free-label game

Now we turn our attention to games with $c^L > 0$. As mentioned earlier, these games are inherently harder than the ones with free labels. For this setting, we use an ϵ-greedy style algorithm, together with discretization.

For this variation of the game, if the learner chooses to see the true label at the end of each round (*i.e.,* $s_t^L = 1$) it suffers an extra loss of $c^L > 0$ in that round.

As in games with free label, here we also use discretization. Then, on the discretized set of actions, we employ an algorithm that is very similar to the algorithm "Random Forecaster with a Revealing Action" [Cesa-Bianchi et al., 2006, Figure 2.].

The idea of the algorithm is that it plays following exponential weights on the elements of $\mathcal{D}_\alpha$. When $w_t \in \mathcal{D}_\alpha$ is selected, only the features that are "needed" are requested, that is, $s_t(i) = \mathbb{I}_{\{w_t(i) \neq 0\}}$. For brevity, we denote this vector $s_t = s(w_t)$. Additionally, at the beginning of each turn, a Bernoulli random variable Z_t is drawn with preset parameter γ and, if $Z_t = 1$ then, the algorithm requests the label and also asks for all the features (that is, $s_t = \mathbf{1}$, $s_t^L = 1$). We will call these rounds *exploration rounds*. The extra loss suffered in these rounds is the cost of the label (c^L) and the cost of features whose $w_t(i)$ coordinate is zero.

In exploration rounds, the losses of all actions can be calculated, and thus the weights of all actions will be updated via importance weighting. Pseudocode for the algorithm can be found in Algorithm 20.

The following theorem is an upper bound on the expected regret achieved

by Algorithm 20.

Theorem 16. *Given any online probing game with costly labels, Algorithm 20 with appropriately set parameters achieves*

$$\mathbb{E}[R_T] \leq CT^{2/3}(\ell_{\max} c_{\max} d \log(TLd))^{1/3}$$

for some constant $C > 0$.

Proof. The regret of the algorithm is decomposed into three additive terms:

1. The approximation error due to discretization. By Lemma 24, we know that the (cumulative) approximation error can be upper bounded by $\frac{TL\sqrt{d}}{\alpha - 1}$.
2. The extra loss suffered in exploration rounds. The cumulative expectation of this extra loss can be upper bounded by $T\gamma(c^L + c^\top \mathbf{1})$.
3. The regret of the algorithm compared to the discretized set of weights, excluding actions that request the label. To upper bound this term, we follow the classical "exponential weights" proof (see *e.g.,* Cesa-Bianchi et al. [2006]).

First we make the trivial observation that for every time step t and weight vector $w \in \mathcal{D}_\alpha, \mathbb{E}[\tilde{\ell}_t(w)] = \ell(w, s(w), y_t)$. That is, $\tilde{\ell}_t(w)$ is an unbiased estimate of the true loss $\ell(w, s(w), y_t)$. Let N denote the number of discrete actions $|\mathcal{D}_\alpha|$. Now we continue with lower and upper bounding the term U_T/U_0:

$$\frac{U_T}{U_0} \geq \frac{\sum_{w \in \mathcal{D}_\alpha} u_T(w)}{N} \geq \frac{u_T(w^*)}{N} = \frac{\exp\left(-\eta \sum_{t=1}^T \tilde{\ell}_t(w^*)\right)}{N}.$$

where w^* denotes an optimal weight vector. For the upper bound we write

$$\begin{aligned}
\frac{U_t}{U_{t-1}} &= \sum_{w \in \mathcal{D}_\alpha} \frac{u_{t-1}(w) \exp(-\eta \tilde{\ell}_t(w))}{U_{t-1}} \\
&= \sum_{w \in \mathcal{D}_\alpha} p_t(w)(1 - \eta \tilde{\ell}_t(w) + \eta^2 \tilde{\ell}_t^2(w)) \qquad (8.1) \\
&= 1 - \eta \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t(w) + \eta^2 \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t^2(w) \\
&\leq \exp\left(-\eta \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t(w) + \eta^2 \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t^2(w)\right), \qquad (8.2)
\end{aligned}$$

where in (8.1) we used that $u_{t-1}(w)/U_{t-1} = p_t(w)$ and the inequality $e^x \leq 1 + x + x^2$ if $x \leq 1$, and in (8.2) we used that $e^x \geq 1 + x$. Multiplying the above inequality for $t = 1, \ldots, T$ we get

$$\frac{U_T}{U_0} \leq \exp\left(-\eta \sum_{t=1}^T \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t(w) + \eta^2 \sum_{t=1}^T \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t^2(w)\right).$$

We now merge the lower and upper bounds and take logarithm of both sides:

$$-\eta \sum_{t=1}^{T} \tilde{\ell}_t(w^*) - \log N \leq -\eta \sum_{t=1}^{T} \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t(w) + \eta^2 \sum_{t=1}^{T} \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t^2(w) \,.$$

Rearranging gives

$$\sum_{t=1}^{T} \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t(w) - \sum_{t=1}^{T} \tilde{\ell}_t(w^*) \leq \eta \sum_{t=1}^{T} \sum_{w \in \mathcal{D}_\alpha} p_t(w) \tilde{\ell}_t^2(w) + \frac{\log N}{\eta} \,.$$

After taking expectation of both sides, the first term on the left hand side is the expected cumulative loss of the algorithm excluding the extra loss suffered in exploration rounds, while the second term is the expected cumulative loss of the best action w^*. The first term on the right hand side can be upper bounded as

$$\begin{aligned} \eta \sum_{t=1}^{T} \sum_{w \in \mathcal{D}_\alpha} \mathbb{E}[p_t(w) \tilde{\ell}_t^2(w)] &\leq \eta \sum_{t=1}^{T} \sum_{w \in \mathcal{D}_\alpha} \mathbb{E}[p_t(w) \tilde{\ell}_t(w)] \frac{\ell_{\max}}{\gamma} \\ &\leq \frac{\eta \ell_{\max} T}{\gamma} \,, \end{aligned}$$

where $\ell_{\max}$ is the maximum loss an action can suffer, ignoring the label cost c^L. Adding up all the three terms of the expected regret, substituting $N = \log \alpha^d$, and denoting $c_{\max} = c^L + c^\top \mathbf{1}$ we get

$$\mathbb{E}[R_T] \leq \frac{TL\sqrt{d}}{\alpha - 1} + T\gamma c_{\max} + \frac{\eta \ell_{\max} T}{\gamma} + \frac{d \log \alpha}{\eta} \,.$$

Setting the parameters to

$$\alpha = TL\sqrt{d} \qquad \eta = (d \log \alpha)^{2/3} T^{-2/3} (\ell_{\max} c_{\max})^{-1/3} \qquad \gamma = \sqrt{\frac{\eta \ell_{\max}}{c_{\max}}}$$

we get

$$\mathbb{E}[R_T] \leq C T^{2/3} (\ell_{\max} c_{\max} d \log(TLd))^{1/3}$$

for some constant $C > 0$. □

8.4 Lower bound for the non-free-label game

In this section we present a lower bound on the expected regret of a non-trivial class of non-free-label games. As we see, this lower bound is within a logarithmic factor of the upper bound from Section 8.3.

Theorem 17. *Let the prediction loss function be $\hat{\ell}(\hat{y}, y) = \|\hat{y} - y\|^2$ (the square loss). There exists a constant C such that for any non-free-label game with $c_j > (1/d)\sum_{i=1}^{d} c_i - 2/d$ for every $j = 1, \ldots, d$, the expected regret of any algorithm can be lower bounded by*

$$\mathbb{E}[R_T] \geq C(c^L d)^{1/3} T^{2/3} .$$

Proof. We construct a set of opponent strategies and show that the expected regret of any algorithm is high against at least one of them. The features $x_{t,i}$ for $t = 1, \ldots, T$ and $i = 1, \ldots, d$ are generated by the iid random variables $X_{t,i}$ whose distribution is Bernoulli with parameter 0.5. Let $Z_t \in \{1, \ldots, d\}$ be random variables whose distribution will be specified later. The labels y_t are generated by the random variable defined as $Y_t = X_{t,Z}$.

To construct the distribution of Z_t we introduce the following notation. For every $i = 1, \ldots, d$, let

$$a_i = \frac{1}{d} + 2c_i - \frac{2}{d}\sum_{j=1}^{d} c_j .$$

The assumptions on c ensures that $a_i > 0$ for every $i = 1, \ldots, d$. For opponent strategy k, let the distribution of Z_t defined as

$$\mathbb{P}_k\left(Z_t = i\right) = \begin{cases} a_i - \epsilon, & i \neq k; \\ a_i + (d-1)\epsilon, & \text{i=k}\,, \end{cases}$$

with some $\epsilon > 0$ to be defined later.

Lemma 25. *Let e_k denote the k^{th} basis vector of dimension d. Against opponent strategy k, the instantaneous expected regret for any action such that $(s, s_\ell) \neq (e_k, 0)$ is at least $\frac{d\epsilon}{2}$.*

For $i = 1, \ldots, d$, let N_i denote the number of times the player's action is (e_i, w, s^L). Similarly, let N_L denote the number of times the player requests the label. Now it is easy to see that the expected regret under opponent strategy k can be lower bounded by

$$\mathbb{E}_k[R_T] \geq \left(T - \mathbb{E}_k[N_k]\right)\frac{d\epsilon}{2} + c^L \mathbb{E}_k[N_L] .$$

The rest of the proof is devoted to show that for any algorithm, the average of the above value, $1/d\sum_{i=1}^{d}\mathbb{E}_i[R_T]$ can be lower bounded. We only show this for deterministic algorithms. The statement follows for randomizing algorithms with the help of a simple argument, see *e.g.,* Cesa-Bianchi and Lugosi [2006, Theorem 6.11].

A deterministic algorithm is defined as a sequence of functions $A_t(\cdot)$, where the argument of A_t is a sequence of observations up to time step $t-1$ and the value is the action taken at time step t. We denote the observation at time step t by $h_t \in \{0, 1, *\}^d$ and $h_t^L \in \{0, 1, *\}$, where $h_{t,i} = x_{t,i}$ if $s_{t,i} = 1$ and

$h_{t,i} = *$ if $s_{t,i} = 0$. Similarly, $h_t^L = y_t$ if $s_t^L = 1$ and $h_t^L = *$ if $s_t^L = 0$. That is, $*$ is the symbol for not observing a feature or the label. The next lemma, which is the key lemma of the proof, shows that the expected value of N_i does not change too much if we change the opponent strategy.

Lemma 26. *There exists a constant C_1 such that for any $i, j \in \{1, \ldots, d\}$,*

$$\mathbb{E}_i[N_i] - \mathbb{E}_j[N_i] \leq C_1 T\epsilon\sqrt{d\mathbb{E}_j[N_L]}\,.$$

Now we are equipped to lower bound the expected regret. Let

$$j = \operatorname{argmin}_{k\in\{1,\ldots,d\}} \mathbb{E}_k[N_L].$$

By Lemma 26,

$$\begin{aligned}\mathbb{E}_i[R_T] &\geq (T - \mathbb{E}_i[N_i])\,\frac{d\epsilon}{2} + c^L\mathbb{E}_i[N_L] \\ &\geq \left(T - \mathbb{E}_j[N_i] - C_1 T\epsilon\sqrt{d\mathbb{E}_j[N_L]}\right)\frac{d\epsilon}{2} + c^L\mathbb{E}_j[N_L]\end{aligned}$$

Denoting $\sqrt{\mathbb{E}_j[N_L]}$ by ν we have

$$\begin{aligned}\frac{1}{d}\sum_{i=1}^{d}\mathbb{E}_i[R_T] &\geq \left(T - \frac{1}{d}\sum_{i=1}^{d}\mathbb{E}_j[N_i] - C_1 T\epsilon\sqrt{d}\nu\right)\frac{d\epsilon}{2} + c^L\nu^2 \\ &\geq \left(T - \frac{T}{d} - C_1 T\epsilon\sqrt{d}\nu\right)\frac{d\epsilon}{2} + c^L\nu^2\end{aligned}$$

What is left is to optimize this bound in terms of ν and ϵ. Since ν is the property of the algorithm, we have to minimize the expression in ν, with ϵ as a parameter. After simple algebra we get

$$\nu_{opt} = \frac{C_1 T\epsilon^2 d^{3/2}}{4c^L}\,.$$

Substituting it back results in

$$\frac{1}{d}\sum_{i=1}^{d}\mathbb{E}_i[R_T] \geq (d-1)\frac{T\epsilon}{2} - \frac{C_1^2 T^2\epsilon^4 d^3}{16c^L}$$

Now we set

$$\epsilon = \left(\frac{2}{C_1^2}\right)^{1/3}\left(c^L\right)^{1/3} d^{-2/3}T^{-1/3}$$

to get

$$\mathbb{E}[R_T] \geq C_3\left(c^L\right)^{1/3} d^{1/3}T^{2/3}$$

whenever $d \geq 2$. □

Chapter 9

Conclusions

In this work we have addressed the problem of minimizing regret in online learning with arbitrary feedback structures, under the framework of partial monitoring. We found that games with finite number of actions and outcomes can be categorized into four classes based on their minimax regret: we distinguished trivial, easy, hard, and hopeless games. These names intuitively express the "hardness" of learning to perform optimally in a specific game. We found that the condition separating easy from hard games is the local observability condition, the condition that enables the learner to accurately estimate the difference of the losses of neighboring actions.

Partial monitoring—apart from being a common generalization of bandit and full-information games—covers many games of interest. In this work we showed that the game of apple tasting belongs to the easy class, while the game of dynamic pricing belongs to the hard class. We also showed that under some reasonable restrictions of the opponent, even in hard games like dynamic pricing, the learner can achieve as low regret as if the game was easy.

To achieve near optimal regret for easy games under various conditions, we developed several new algorithms. What is common in almost all of them is that instead of estimating the losses of each action, they estimate *loss differences*. Estimating loss differences instead of losses is just as sufficient for the purpose of minimizing regret. On the other hand, in many cases the estimate of the difference can be more accurate than the estimate of the loss, and thus algorithms that rely on loss difference estimates can achieve better regret. We believe that this finding can lead to better algorithms in many fields of machine learning in the future.

As for future work, a lot remains to be done. Here we mention only a few topics. First, it remains to extend our investigation to games with infinitely many actions and/or outcomes, and games with a few basic actions but a large number of *experts* whose actions at any time step can be one of the primitive actions. Another important open question is to analyze partial monitoring with side information when in every round, the learner receives some additional side information about the choice of the opponent before choosing his action. Then, the learner's performance is compared with that of the best "policy": a

function that maps side information to actions. The hope is that the advantage of locally observable games is preserved, that is, if a game is locally observable then its minimax regret scales as $\widetilde{\Theta}(\sqrt{T})$ even if we add side information to the game.

Bibliography

Y. Abbasi-Yadkori, D. Pál, and Cs. Szepesvári. Improved algorithms for linear stochastic bandits (extended version). In *NIPS*, December 2011.

J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, volume 3, 2008.

R. Agrawal. Sample mean based index policies with O (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

C. Allenberg, P. Auer, L. Györfi, and Gy. Ottucsák. Hannan consistency in on-line learning in case of unbounded losses under partial monitoring. In *ALT2006*, 2006.

A. Antos, G. Bartók, D. Pál, and Cs. Szepesvári. Toward a classification of finite partial-monitoring games. *Theoretical Computer Science*, 2012. to appear.

J.-Y. Audibert and S. Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11:2785–2836, 2010.

P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:422, 2003.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.*, 47(2-3):235–256, 2002. ISSN 0885-6125.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

P. Auer, R. Ortner, and Cs. Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. *Learning Theory*, pages 454–468, 2007.

G. Bartók, D. Pál, and Cs. Szepesvári. Toward a classification of finite partial-monitoring games. In *ALT*, pages 224–238, 2010.

G. Bartók, D. Pál, and Cs. Szepesvári. Minimax regret of finite partial-monitoring games in stochastic environments. In *COLT 2011, Proceedings of the 24th Annual Conference on Learning Theory, Budapest, Hungary, July 9–11, 2011*, 2011.

G. Bartók, N. Zolghadr, and Cs. Szepesvári. An adaptive algorithm for finite stochastic partial monitoring. In *ICML*, 2012. submitted.

D. Blackwell. Controlled random walks. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 336–338, 1954.

S. Bubeck, R. Munos, G. Stoltz, and Cs. Szepesvari. Online optimization in $\mathcal{X}$–armed bandits. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 201–208, 2009.

S. Bubeck, R. Munos, G. Stoltz, and Cs. Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, June 2011. Submitted on 21/1/2010.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge Univ Pr, 2006.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. In *IEEE Transactions on Information Theory*, volume 50, pages 2050–2057, 2004.

N. Cesa-Bianchi, G. Lugosi, and G. Stoltz. Regret minimization under partial monitoring. *Math. Oper. Res.*, 31(3):562–580, 2006.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, New York, second edition, 2006.

I. Csiszár and J. Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, New York, 1981.

V. Dani, T.P. Hayes, and S.M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21th Annual Conference on Learning Theory (COLT 2008)*, pages 355–366. Citeseer, 2008.

O. Dekel and Y. Singer. Data-driven online to batch conversions. In *NIPS'05*, 2006.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Applications of Mathematics: Stochastic Modelling and Applied Probability. Springer-Verlag New York, 1996.

D.P. Foster and A. Rakhlin. No internal regret via neighborhood watch. *CoRR*, abs/1108.6088, 2011.

J. Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *COLT*, pages 57–68, 2008.

E. Hazan and S. Kale. Better algorithms for benign bandits. *Journal of Machine Learning Research*, 12:1287–1311, 2011.

D.P. Helmbold, N. Littlestone, and P.M. Long. Apple tasting. *Information and Computation*, 161(2):85–139, 2000.

R. Kleinberg, A. Slivkins, and E. Upfal. Multi-armed bandits in metric spaces. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 681–690, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-047-0.

T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

N Littlestone. From on-line to batch learning. In *COLT*, pages 269–284, 1989.

N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

J.E. Littlewood. On bounded bilinear forms in an infinite number of variables. *The Quarterly Journal of Mathematics*, 1:164–174, 1930.

S. Mannor and O. Shamir. From bandits to experts: On the value of side-observations. In *NIPS*, 2011.

H.B. McMahan and M. Streeter. Tighter bounds for multi-armed bandits with expert advice. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009.

V. Mnih. Efficient stopping rules. Master's thesis, Department of Computing Science, University of Alberta, 2008.

V. Mnih, Cs. Szepesvári, and J.-Y. Audibert. Empirical Bernstein stopping. In *ICML*, pages 672–679, 2008.

A. Piccolboni and C. Schindelhauer. Discrete prediction games with arbitrary feedback and loss. *Lecture notes in computer science*, pages 208–223, 2001.

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 55:527–535, 1952.

P. Rusmevichientong and J.N. Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395, 2010.

V.G. Vovk. Aggregating strategies. In *Annual Workshop on Computational Learning Theory: Proceedings of the third annual workshop on Computational learning theory*. Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA,, 1990.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 928–936, 2003.

Appendix A

Proofs of the lemmas

A.1 Lemmas from Chapter 4

Lemma 1. *For any finite partial-monitoring game, the following four statements are equivalent:*

a) The minimax regret is zero for each T.

b) The minimax regret is zero for some T.

c) There exists a (non-dominated) action $i \in \underline{N}$ whose loss is not larger than the loss of any other action irrespectively of the choice of opponent's action.

d) The number of non-dominated actions is one ($K = 1$).

Proof. a)→b) is obvious.
b)→c) For any $\mathcal{A}$,

$$\begin{aligned}\mathbb{E}[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] &\geq \sup_{j\in\underline{M},J_1=\cdots=J_T=j} \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{L}[I_t,J_t] - \min_{i\in\underline{N}}\sum_{t=1}^{T}\mathbf{L}[i,J_t]\right]\\ &= \sup_{j\in\underline{M}} \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{L}[I_t,j] - T\min_{i\in\underline{N}}\mathbf{L}[i,j]\right]\\ &\geq \sup_{j\in\underline{M}}\left(\mathbb{E}\left[\mathbf{L}[I_1,j]\right] - \min_{i\in\underline{N}}\mathbf{L}[i,j]\right) = f(\mathcal{A})\ .\end{aligned}$$

b) leads to

$$0 = \mathrm{R}_T(\mathbf{G}) = \inf_{\mathcal{A}}\mathbb{E}[\mathrm{R}_T^{\mathcal{A}}(\mathbf{G})] \geq \inf_{\mathcal{A}} f(\mathcal{A})\ .$$

Observe that $f(\mathcal{A})$ depends on $\mathcal{A}$ through only the distribution of I_1 on $\underline{N}$ denoted by $q = q(\mathcal{A})$ now, that is, $f(\mathcal{A}) = f'(q)$ for proper f'. This dependence is continuous on the compact domain of q, hence the infimum can be replaced by minimum. Thus $\min_q f'(q) \leq 0$, that is, there exists a q such that for all $j \in \underline{M}$, $\mathbb{E}\left[\mathbf{L}[I_1,j]\right] = \min_{i\in\underline{N}}\mathbf{L}[i,j]$. This implies that the support of

q contains only actions whose loss is not larger than the loss of any other action irrespectively of the choice of the opponent's action. (Such an action is obviously non-dominated as shown by any $p \in \Delta_M$ supported on all outcomes.)

c)→d) Action i in c) is non-dominated, and any other action with loss vector distinct from ℓ_i is dominated (by i and any action with loss vector ℓ_i).

d)→a) Since there is only one non-dominated action, an algorithm that chooses that action in every time step suffers zero regret. □

Lemma 2. *Let S be the number of times* AppleTree *calls* Reset *at the root node. Then there exists a universal constant c^* such that $S \le \frac{c^* \ln T}{\Delta}$, where $\Delta = \rho'_2 - \rho'_1$ with ρ'_1 and ρ'_2 given by* (4.2).

Proof. Let s be the number of times the algorithm switches from $\mathbf{G}_2$ to $\mathbf{G}_1$. Let $t_1 < \cdots < t_s$ be the time steps t when g_t switches from 2 to 1, i.e., when $\hat{\rho}_t < \rho'_1$ and $g_{t-1} = 2$ (and thus $g_t = 1$). Similarly, let $t'_1 < \cdots < t'_{s+\xi}$, $(\xi \in \{0,1\})$ be the time steps t when g_t switches from 1 to 2, i.e., when $\hat{\rho}_t > \rho'_2$ and $g_{t-1} = 1$ (and thus $g_t = 2$). Note that for all $1 \le j < s$, $t'_j < t_j < t'_{j+1}$. Finally, for every $1 \le j < s$, we define t''_j to be the time step $t \ge t'_j$ when $\hat{\rho}_t$ drops below 1 and then stays there until the next reset: $t''_j = \min\{t \mid t'_j \le t \le t_j, \forall \tau \in \{t, t+1, \ldots, t_j\}, \hat{\rho}_\tau \le 1\}$.

First, we observe that if $t''_j \ge 2/\Delta$ then $\hat{\rho}_{t''_j} \ge (\rho'_1 + \rho'_2)/2$. Indeed, if $t''_j = t'_j$ then $\hat{\rho}_{t''_j} \ge \rho'_2$, while if $t''_j \ne t'_j$ then $\hat{\rho}_{t''_j - 1} > 1$ and thus, from the update rule, we have

$$\hat{\rho}_{t''_j} = \left(1 - \frac{1}{t''_j}\right) \hat{\rho}_{t''_j - 1} + \frac{1}{t''_j} \cdot \frac{\mathbb{I}_{\{J_{t''_j} = 2\}}}{p_{t''_j}} \ge 1 - \frac{\Delta}{2} \ge \frac{\rho'_1 + \rho'_2}{2} \,.$$

The number of times the algorithm resets is at most $2s + 1$. Let j^* be the first index such that $t''_{j^*} \ge 2/\Delta$. Pick any j such that $j^* \le j \le s$. According to the update rule, for any $t''_j < t \le t_j$ we have that

$$\hat{\rho}_t = \left(1 - \frac{1}{t}\right) \hat{\rho}_{t-1} + \frac{1}{t} \cdot \frac{\mathbb{I}_{\{J_t = 2\}}}{p_t} \ge \hat{\rho}_{t-1} - \frac{1}{t}\hat{\rho}_{t-1} \ge \hat{\rho}_{t-1} - \frac{1}{t}$$

and hence $\hat{\rho}_{t-1} - \hat{\rho}_t \le \frac{1}{t}$. Summing this inequality for $t = t''_j + 1, \ldots, t_j$ and exploiting that $\hat{\rho}_{t''_j} \ge (\rho'_1 + \rho'_2)/2$ and $\hat{\rho}_{t_j} \le \rho'_1$, we get

$$\frac{\Delta}{2} = \frac{\rho'_1 + \rho'_2}{2} - \rho'_1 \le \hat{\rho}_{t''_j} - \hat{\rho}_{t_j} \le \sum_{t = t''_j + 1}^{t_j} \frac{1}{t} = O\left(\ln \frac{t_j}{t''_j}\right) \,.$$

Thus, there exists $c > 0$ such that for all $j^* \le j \le s$, it holds that

$$\frac{1}{c}\Delta \le \ln \frac{t_j}{t''_j} \le \ln \frac{t_j}{t_{j-1}} \,. \tag{A.1}$$

Summing up (A.1) for $j = j^*, \ldots, s$, we get $(s - j^*)\frac{1}{c}\Delta \le \ln \frac{t_s}{2/\Delta} \le \ln T$. We conclude the proof by observing that $j^* \le 2/\Delta$. □

Lemma 3. *For any $0 < \delta < 1$, with probability at least $1 - \delta$, for all $t \geq 8\sqrt{T}\ln(2T/\delta)/(3\Delta^2)$, $|\hat{\rho}_t - \rho_t| \leq \Delta$.*

Proof. The proof of the lemma employs Bernstein's inequality for martingales.

Bernstein's Inequality for Martingales. *(Taken from Cesa-Bianchi and Lugosi [2006, Lemma A.8]) Let $X_1, X_2, \ldots, X_n$ be a bounded martingale difference sequence with respect to a filtration $\{\mathcal{F}\}_{i=0}^n$ and with $|X_i| \leq K$. Let*

$$S_i = \sum_{j=1}^{i} X_j$$

be the associated martingale. Denote the sum of conditional variances by

$$\Sigma_n^2 = \sum_{i=1}^{n} \mathbb{E}[X_i^2 \mid \mathcal{F}_{i-1}] \ .$$

Then, for all constants $\epsilon, v > 0$,

$$\mathbb{P}\left(\max_{i \in \underline{n}} S_i > \epsilon \text{ and } \Sigma_n^2 \leq v\right) \leq \exp\left(-\frac{\epsilon^2}{2(v + K\epsilon/3)}\right) \ .$$

For $1 \leq t \leq T$, let p_t be the conditional probability of playing a revealing action at time step t, given the history $\mathcal{H}_{1:t-1}$. Recall that, due to the construction of the algorithm, $p_t \geq 1/\sqrt{T}$.

If we write $\hat{\rho}_t$ in its explicit form $\hat{\rho}_t = \frac{1}{t}\sum_{s=1}^{t} \frac{\mathbb{I}_{\{H_s=2\}}}{p_s}$ we can observe that $\mathbb{E}[\hat{\rho}_t|\mathcal{H}_{1:t-1}] = \rho_t$, that is, $\hat{\rho}_t$ is an unbiased estimate of the relative frequency. Let us define random variables $X_s := \frac{\mathbb{I}_{\{H_s=2\}}}{p_s} - \mathbb{I}_{\{J_s=2\}}$. Since p_s is determined by the history, $\{X_s\}_s$ is a martingale difference sequence. Also, from $p_s \geq 1/\sqrt{T}$ we know that $\mathbb{V}ar[(]X_s|\mathcal{H}_{1:t-1}) \leq \sqrt{T}$. Hence, we can use Bernstein's inequality for martingales with $\epsilon = \Delta t$, $\nu = t\sqrt{T}$, $K = \sqrt{T}$:

$$\begin{aligned}
\mathbb{P}\left(|\hat{\rho}_t - \rho_t| > \Delta\right) &= \mathbb{P}\left(\left|\sum_{s=1}^{t} X_s\right| > t\Delta\right) \\
&\leq 2\exp\left(-\frac{\Delta^2 t^2/2}{t\sqrt{T} + \Delta t\sqrt{T}/3}\right) \\
&\leq 2\exp\left(-\frac{3\Delta^2 t}{8\sqrt{T}}\right) \ .
\end{aligned}$$

We have that if $t \geq 8\sqrt{T}\ln(2T/\delta)/(3\Delta^2)$ then

$$\mathbb{P}\left(|\hat{\rho}_t - \rho_t| > \Delta\right) \leq \delta/T \ .$$

We get the bound for all $t \in [8\sqrt{T}\ln(2T/\delta)/(3\Delta^2), T]$ using the union bound. □

Lemma 4. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be any finite non-trivial game with N actions and $M \geq 2$ outcomes. Then there exists $p \in \Delta_M$ satisfying both of the following properties:*

(a) All coordinates of p are positive.

(b) There exist actions i_1,$i_2 \in \underline{N}$ such that $\ell_{i_1} \neq \ell_{i_2}$ and for all $i \in \underline{N}$,

$$\ell_{i_1}^\top p = \ell_{i_2}^\top p \leq \ell_i^\top p \; .$$

Proof. Note that distributions p with positive coordinates form the interior of Δ_M ($\operatorname{Int} \Delta_M$). For any action $i \in \underline{N}$, as in the proof of Lemma 6, consider the compact convex cell C_i in Δ_M, whose union is Δ_M (see (A.2)). Let p_1 be any point in the interior of Δ_M. By (A.2), there is a cell C_{i_1} containing p_1. If $C_{i_1} = \Delta_M$ held then action i_1 would satisfy Lemma 1 c), thus also d), and the game would be trivial. So there must be a point, say p_2, in $\Delta_M \setminus C_{i_1}$. The intersection of the closed segment $\overline{p_1 p_2}$ and C_{i_1} is closed and convex, thus it is a closed subsegment $\overline{p_1 p}$ for some $p \in C_{i_1}$ ($p \neq p_2$). $p_1 \in \operatorname{Int} \Delta_M$ and the convexity of Δ_M imply $p \in \operatorname{Int} \Delta_M$. Since the open segment $\overline{p p_2}$ has to be covered by $\bigcup_{i' : C_{i'} \neq C_{i_1}} C_{i'}$, that is a closed set, $p \in \bigcup_{i : C_{i'} \neq C_{i_1}} C_{i'}$ must also hold, that is, $p \in C_{i_2}$ for some $C_{i_2} \neq C_{i_1}$ (requiring $\ell_{i_1} \neq \ell_{i_2}$). Hence p satisfies both (a) and (b). □

Lemma 5 (Khinchine's inequality for asymmetric random variables)**.** *Let*

$$X_1, X_2, \ldots, X_T$$

be i.i.d. random variables with mean $\mathbb{E}[X_t] = 0$, finite variance $\mathbb{E}[X_t^2] = \mathbb{V}ar[X_t] = \sigma^2$, and finite fourth moment $\mathbb{E}[X_t^4] = \mu_4$. Then,

$$\mathbb{E}\left|\sum_{t=1}^{T} X_t\right| \geq \frac{\sigma^3}{\sqrt{3\mu_4}}\sqrt{T} \; .$$

Proof. [Devroye et al., 1996, Lemma A.4] implies that for any random variable Z with finite fourth moment

$$\mathbb{E}\,|Z| \geq \frac{\left(\mathbb{E}[Z^2]\right)^{3/2}}{\left(\mathbb{E}[Z^4]\right)^{1/2}} \; .$$

Applying this inequality to $Z = \sum_{t=1}^{T} X_t$ we get

$$\mathbb{E}\left|\sum_{t=1}^{T} X_t\right| \geq \frac{T^{3/2}\sigma^3}{T\sqrt{3\mu_4}} = \frac{\sigma^3}{\sqrt{3\mu_4}}\sqrt{T},$$

that follows from

$$\mathbb{E}[Z^2] = \mathbb{E}\left[\left(\sum_{t=1}^{T} X_t\right)^2\right] = \sum_{t=1}^{T} \mathbb{E}[X_t^2] = T\sigma^2$$

and

$$\begin{aligned}\mathbb{E}[Z^4] = \mathbb{E}\left[\left(\sum_{t=1}^{T} X_t\right)^4\right] &= \sum_{t=1}^{T} \mathbb{E}[X_t^4] + 6\sum_{1\le s<t\le T} \mathbb{E}[X_s^2]\,\mathbb{E}[X_t^2] \\ &= T\mu_4 + 3T(T-1)\sigma^4 \\ &\le 3T^2\mu_4,\end{aligned}$$

where we have used the independence of X_t's and $\mathbb{E}[X_t] = 0$ which ensure that mixed terms $\mathbb{E}[X_t X_s]$, $\mathbb{E}[X_t X_s^3]$, etc. vanish. We also used that $\sigma^4 = \mathbb{E}[X_t^2]^2 \le \mathbb{E}[X_t^4] = \mu_4$. □

Lemma 6 (ϵ-close distributions)**.** *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be any finite non-trivial game with N non-duplicate actions and $M \ge 2$ outcomes. Then there exist two non-dominated actions i_1,$i_2 \in \underline{N}$, $p \in \Delta_M$, $w \in \mathbb{R}^M \setminus \{0\}$, and c,$\alpha > 0$ satisfying the following properties:*

(a) $\ell_{i_1} \ne \ell_{i_2}$.

(b) $\ell_{i_1}^\top p = \ell_{i_2}^\top p \le \ell_i^\top p$ *for all* $i \in \underline{N}$ *and the coordinates of* p *are positive.*

(c) Coordinates of w *satisfy* $\sum_{j=1}^{M} w(j) = 0$.

For any $\epsilon \in (0, \alpha)$,

(d) $p_1 = p + \epsilon w \in \Delta_M$ *and* $p_2 = p - \epsilon w \in \Delta_M$,

(e) for any $i \in \underline{N}$, $i \ne i_1$, *we have* $(\ell_i - \ell_{i_1})^\top p_1 \ge c\epsilon$,

(f) for any $i \in \underline{N}$, $i \ne i_2$, *we have* $(\ell_i - \ell_{i_2})^\top p_2 \ge c\epsilon$.

Proof. For any action $i \in \underline{N}$, consider the cell

$$C_i = \{p \in \Delta_M \ : \ \forall i' \in \underline{N},\ \ell_i^\top p \le \ell_{i'}^\top p\}$$

in the probability simplex Δ_M. The cell C_i corresponds to the set of outcome distributions under which action i is optimal. Each cell is the intersection of some closed half-spaces and Δ_M, and thus it is a compact convex polytope of dimension at most $M - 1$. Note that

$$\bigcup_{i=1}^{N} C_i = \Delta_M. \tag{A.2}$$

For $C \subseteq \Delta_M$, denote $\operatorname{Int} C$ its interior in the topology induced by the hyperplane $\{x \in \mathbb{R}^M \ : \ (1, dots, 1)x = 1\}$ and $\operatorname{rint} C$ its relative interior[1]. Let λ be

[1] *Relative interior* of $C \subseteq \mathbb{R}^M$ is its interior in the topology induced by the smallest affine space containing it.

the $(M-1)$-dimensional Lebesgue-measure. It is easy to see that for any pair of cells C_i, $C_{i'}$, $C_{i'} \cap \operatorname{Int} C_i = \emptyset$, that is, $\lambda(C_i \cap C_{i'}) = 0$, and so

$$\operatorname{Int} C_i \subseteq C_i \setminus \bigcup_{i' \neq i} C_{i'}. \tag{A.3}$$

Hence the cells form a cell-decomposition of the simplex. Any two cells C_i and $C_{i'}$ are separated by the hyperplane $f_{i,i'} = \{x \in \mathbb{R}^M \ : \ \ell_i^\top x = \ell_{i'}^\top x\}$. Note that $C_i \cap C_{i'} \subset f_{i,i'}$. The cells are characterized by the following lemma (which itself holds also with duplicate actions):

Lemma 27. *Action i is dominated $\Leftrightarrow$ $C_i \subseteq \bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'}$ $\Leftrightarrow$ $\operatorname{Int} C_i = \emptyset$ $\Leftrightarrow$ $\lambda(C_i) = 0$, that is, C_i is $(M-1)$-dimensional (has positive λ-measure) if and only if there is $p \in C_i \setminus \bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'}$. Hence there is three kind of "cells":*

1. *$C_i = \emptyset$ (action i is never optimal),*
2. *$C_i \neq \emptyset$ has dimension less than $M-1$, $\operatorname{Int} C_i = \emptyset$, $\lambda(C_i) = 0$, $C_i \subseteq \bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'}$ (action i is degenerate),*
3. *action i is non-dominated, C_i is $(M-1)$-dimensional, $\operatorname{rint} C_i = \operatorname{Int} C_i \neq \emptyset$, $\lambda(C_i) > 0$, there is $p \in C_i \setminus \bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'}$.*

Moreover $\bigcup_{i \notin \mathcal{D}} C_i = \Delta_M$ for the set $\mathcal{D}$ of dominated actions.[2]

Proof of Lemma 27. By Definition 3, action i is dominated if and only if $C_i \subseteq \bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'}$.

$C_i \subseteq \bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'} \to \operatorname{Int} C_i = \emptyset$: Since $\ell_{i'} \neq \ell_i \to i \neq i$, follows from (A.3).

$\operatorname{Int} C_i = \emptyset \to \lambda(C_i) = 0$: Follows from convexity of C_i.

$\lambda(C_i) = 0 \to C_i \subseteq \bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'}$: indirect: if $p \in C_i$ is in the complementer of $\bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'}$, that is open in Δ_M, then there is a neighborhood S of p in Δ_M disjoint from $\bigcup_{i':\ell_{i'} \neq \ell_i} C_{i'}$. Thus $S \subseteq \bigcup_{i':\ell_{i'} = \ell_i} C_{i'} = C_i$ due to (A.2), and $\lambda(C_i) \geq \lambda(S) > 0$, contradiction.

Since $\lambda(\bigcup_{i \in \mathcal{D}} C_i) \leq \sum_{i \in \mathcal{D}} \lambda(C_i) = 0$, thus from (A.2) $\lambda(\bigcup_{i \notin \mathcal{D}} C_i) \geq \lambda(\Delta_M)$, and $\lambda(\Delta_M \setminus \bigcup_{i \notin \mathcal{D}} C_i) = 0$. The latest set is open in Δ_M, so it must be empty, that is, $\bigcup_{i \notin \mathcal{D}} C_i = \Delta_M$. □

The non-triviality of the game ($K \geq 2$) means that there are at least two non-dominated actions of type 3 above. In the cell decomposition, due to Lemma 27, there must exist two such $(M-1)$-dimensional cells C_{i_1} and C_{i_2} corresponding to two non-dominated actions i_1,i_2, such that their intersection $C_{i_1} \cap C_{i_2}$ is an $(M-2)$-dimensional polytope. Clearly, $\ell_{i_1} \neq \ell_{i_2}$, since otherwise the cells would coincide; thus part (a) is satisfied.

Moreover, $\operatorname{rint}(C_{i_1} \cap C_{i_2}) \subseteq \operatorname{rint} \Delta_M$ since otherwise $\lambda(C_{i_1})$ or $\lambda(C_{i_2})$ would be zero. We can choose any $p \in \operatorname{rint}(C_{i_1} \cap C_{i_2})$. This choice of p guarantees that $p \in f_{i_1,i_2}$, $\ell_{i_1}^\top p = \ell_{i_2}^\top p$, $p \in \operatorname{rint} \Delta_M$, and part (b) is satisfied. Since $C_{i_1} \cap C_{i_2}$

[2]This last statement is just Lemma 5 in [Piccolboni and Schindelhauer, 2001].

is $(M-2)$-dimensional, it also implies that there exists $\delta > 0$ such that the δ-neighborhood $\{q \in \mathbb{R}^M \ : \ \|p-q\| < \delta\}$ of p is contained in $\operatorname{rint}(C_{i_1} \cup C_{i_2})$.

Since $p \in f_{i_1,i_2}$ therefore the hyperplane of vectors satisfying (c) does not coincide with f_{i_1,i_2} implying that we can choose $w \in \mathbb{R}^M \setminus \{0\}$ satisfying part (c), $\|w\| < \delta$, and $w \notin f_{i_1,i_2}$. We can assume

$$(\ell_{i_2} - \ell_{i_1})^\top w > 0 \tag{A.4}$$

(otherwise we choose $-w$). Since $p \pm w$ lie in the δ-neighborhood of p, they lie in $\operatorname{rint}(C_{i_1} \cup C_{i_2})$. In particular, since $\ell_{i_1}^\top(p+w) < \ell_{i_2}^\top(p+w)$ and $\ell_{i_2}^\top(p-w) < \ell_{i_1}^\top(p-w)$, $p+w \in \operatorname{rint} C_{i_1}$ and $p-w \in \operatorname{rint} C_{i_2}$. Let

$$p_1 = p + \epsilon w \qquad \text{and} \qquad p_2 = p - \epsilon w \ . \tag{A.5}$$

The convexity of C_{i_1} and C_{i_2} implies that for any $\epsilon \in (0,1]$, $p_1 \in \operatorname{rint} C_{i_1}$ and $p_2 \in \operatorname{rint} C_{i_2}$. This, in particular, ensures that p_1,$p_2 \in \Delta_M$ and part (d) holds.

To prove (e) define $\mathcal{I} = \{i \in \underline{N} : \ell_i \text{ is collinear with } \ell_{i_1} \text{ and } \ell_{i_2}\}$. We consider two cases: As the first case fix action $i \in \mathcal{I} \setminus \{i_1\}$, that is, ℓ_i is an affine combination $\ell_i = a_i\ell_{i_1} + b_i\ell_{i_2}$ for some $a_i + b_i = 1$. Since i_1 and i_2 are non-dominated, this must be a convex combination with a_i,$b_i \geq 0$. There is no duplicate action, thus $\ell_i \neq \ell_{i_1}$ implying $b_i \neq 0$. Hence $b_i > 0$, and from (A.5) for any $\epsilon \geq 0$

$$(\ell_i - \ell_{i_1})^\top p_1 = (b_i\ell_{i_2} - b_i\ell_{i_1})^\top(p + \epsilon w) = \epsilon b_i(\ell_{i_2} - \ell_{i_1})^\top w \geq c\epsilon$$

provided that $0 < c \leq \min_{i \in \mathcal{I}\setminus\{i_1\}} b_i(\ell_{i_2} - \ell_{i_1})^\top w = c'$. From (A.4) we know that $b_i(\ell_{i_2} - \ell_{i_1})^\top w$ and so c' are positive.

As the second case suppose $i \notin \mathcal{I}$. Then, the hyperplane $f_{i_1,i}$ does not coincide with f_{i_1,i_2}. Since $p \in \operatorname{rint}(C_{i_1} \cap C_{i_2})$, $p \in f_{i_1,i}$ would contradict to $f_{i_1,i} \cap \operatorname{rint} C_{i_1} = \emptyset$ implied by (A.3). Thus $p \in C_{i_1} \setminus f_{i_1,i}$ and therefore $\ell_{i_1}^\top p < \ell_i^\top p$. This means that if we choose $0 < c \leq \min(c', \frac{1}{2}\min_{i \notin \mathcal{I}}(\ell_i - \ell_{i_1})^\top p)$ (that is positive and depends only on $\mathbf{L}$ and not on T) then for $\epsilon < \alpha = \min(1, c/\max_{i \notin \mathcal{I}} |(\ell_i - \ell_{i_1})^\top w|)$, from (A.5) we have again

$$(\ell_i - \ell_{i_1})^\top p_1 \geq 2c + \epsilon(\ell_i - \ell_{i_1})^\top w > c > c\epsilon \ .$$

Part (f) is proved analogously to part (e), and by adjusting α and c if necessary. □

Lemma 7 (KL divergence of ϵ-close distributions)**.** *Let $p \in \Delta_M$ be a probability vector and let $\underline{p} = \min_{j \in \underline{M}: p(j) > 0} p(j)$. For any vector $\epsilon \in \mathbb{R}^M$ such that both $p - \epsilon$ and $p + \epsilon$ lie in Δ_M and $|\epsilon(j)| \leq p(j)/2$ for all $j \in \underline{M}$, the KL divergence of $p - \epsilon$ and $p + \epsilon$ satisfies*

$$D(p - \epsilon \parallel p + \epsilon) \leq c\|\epsilon\|^2,$$

where $c = \frac{6\ln(3)-4}{\underline{p}} > 0$.

Proof. Since p, $p+\epsilon$, and $p-\epsilon$ are all probability vectors, notice that the coordinates of ϵ have to sum up to zero. Also if a coordinate of p is zero then the corresponding coordinate of ϵ has to be zero as well. As zero coordinates do not modify the KL divergence, we can assume without loss of generality that all coordinates of p are positive. By definition,

$$D(p-\epsilon \parallel p+\epsilon) = \sum_{j=1}^{M} (p(j)-\epsilon(j)) \ln\left(\frac{p(j)-\epsilon(j)}{p(j)+\epsilon(j)}\right) .$$

We write the logarithmic factor as

$$\ln\left(\frac{p(j)-\epsilon(j)}{p(j)+\epsilon(j)}\right) = \ln\left(1-\frac{\epsilon(j)}{p(j)}\right) - \ln\left(1+\frac{\epsilon(j)}{p(j)}\right) .$$

We use the second order Taylor expansion $\ln(1\pm x) = \pm x - x^2/2 + O(|x|^3)$ around 0 to get that $\ln(1-x)-\ln(1+x) = -2x+r(x)$, where $r(x)$ is a remainder upper bounded for all $|x| \le 1/2$ as $|r(x)| \le c'|x|^3$ with $c' = 8\ln(3) - 8 \approx 0.79$. Substituting

$$\begin{aligned} D(p-\epsilon \parallel p+\epsilon) &= \sum_{j=1}^{M} (p(j)-\epsilon(j)) \left[-2\frac{\epsilon(j)}{p(j)} + r\left(\frac{\epsilon(j)}{p(j)}\right)\right] \\ &= -2\sum_{j=1}^{M} \epsilon(j) + 2\sum_{j=1}^{M} \frac{\epsilon^2(j)}{p(j)} + \sum_{j=1}^{M} (p(j)-\epsilon(j)) \cdot r\left(\frac{\epsilon(j)}{p(j)}\right) . \end{aligned}$$

Here the first term is 0. Letting $\underline{p} = \min_{j\in\underline{M}} p(j)$, the second term is bounded by $2\sum_{j=1}^{M} \epsilon^2(j)/\underline{p} = (2/\underline{p})\|\epsilon\|^2$, and the third term is bounded by

$$\begin{aligned} \sum_{j=1}^{M} (p(j)-\epsilon(j)) \left| r\left(\frac{\epsilon(j)}{p(j)}\right)\right| &\le c' \sum_{j=1}^{M} (p(j)-\epsilon(j)) \frac{|\epsilon(j)|^3}{p^3(j)} \\ &= c' \sum_{j=1}^{M} \left(\frac{|\epsilon(j)|}{p(j)} - \frac{\epsilon(j)|\epsilon(j)|}{p^2(j)}\right) \frac{\epsilon^2(j)}{p(j)} \\ &\le c' \sum_{j=1}^{M} \left(\frac{|\epsilon(j)|}{p(j)} + \frac{|\epsilon(j)|^2}{p^2(j)}\right) \frac{\epsilon^2(j)}{p(j)} \\ &\le c' \sum_{j=1}^{M} \left(\frac{1}{2}+\frac{1}{4}\right) \frac{\epsilon^2(j)}{\underline{p}} = \frac{3c'}{4\underline{p}} \|\epsilon\|^2 . \end{aligned}$$

Hence, $D(p-\epsilon \parallel p+\epsilon) \le \frac{8+3c'}{4\underline{p}}\|\epsilon\|^2 = c\|\epsilon\|^2$ for $c = \frac{6\ln(3)-4}{\underline{p}}$. □

Lemma 8. *For any partial-monitoring game with N actions and M outcomes, algorithm $\mathcal{A}$, pair of outcome distributions p_1,$p_2 \in \Delta_M$ and action i, we have*

$$N_i^{(2)} - N_i^{(1)} \leq T\sqrt{D(p_2 \parallel p_1)N_{\text{rev}}^{(2)}/2}$$

and

$$N_i^{(1)} - N_i^{(2)} \leq T\sqrt{D(p_1 \parallel p_2)N_{\text{rev}}^{(1)}/2},$$

where $N_{\text{rev}}^{(k)} = \sum_{t=1}^{T} \mathbb{P}_k\left(I_t \in \mathcal{R}\right) = \sum_{i' \in \mathcal{R}} N_{i'}^{(k)}$ under model p_k, $k = 1,2$ with $\mathcal{R}$ being the set of revealing actions.[3]

Proof. We only prove the first inequality, the other one is symmetric. Assume first that $\mathcal{A}$ is deterministic, that is, $I_t : \Sigma^{t-1} \to \underline{N}$, and so $I_t(h_{1:t-1})$ denotes the choice of the algorithm at time step t, given that the (random) history of observations of length $t-1$, $H_{1:t-1} = (H_1, \ldots, H_{t-1})$ takes $h_{1:t-1} = (h_1, \ldots, h_{t-1}) \in \Sigma^{t-1}$. (Note that this is a slightly different history definition than $\mathcal{H}_{1:t-1}$ defined in Section 4.4.1, as $H_{1:t-1}$ does not include the actions since their choices are determined by the feedback anyway. In general, $\mathcal{H}_{1:t-1}$ is equivalent to $H_{1:t-1} \cup (I_1, ..., I_{t-1})$. Nevertheless, if it is assumed that the feedback symbol sets of actions are disjoint then $H_{1:t-1}$ and $\mathcal{H}_{1:t-1}$ are equivalent.) We denote by p_k^* the joint distribution of $H_{1:T-1}$ over Σ^{T-1} associated with p_k. (For games with only all-revealing actions, assuming $\mathbf{H}[i', j] = j$ in $\mathbf{H}$, p_k^* is the product distribution over the outcome sequences, that is, formally, $p_k^*(j_{1:T-1}) = \prod_{t=1}^{T-1} p_k(j_t)$.) We can bound the difference $N_2^{(2)} - N_2^{(1)}$ as

$$\begin{aligned}
N_i^{(2)} - N_i^{(1)} &= \sum_{t=1}^{T} \left(\mathbb{P}_2\left(I_t = i\right) - \mathbb{P}_1\left(I_t = i\right)\right) \\
&= \sum_{h_{1:T-1} \in \Sigma^{T-1}} \sum_{t=1}^{T} \left(\mathbb{I}_{\{I_t(h_{1:t-1})=i\}} p_2^*(h_{1:T-1}) - \mathbb{I}_{\{I_t(h_{1:t-1})=i\}} p_1^*(h_{1:T-1})\right) \\
&= \sum_{h_{1:T-1} \in \Sigma^{T-1}} \left(p_2^*(h_{1:T-1}) - p_1^*(h_{1:T-1})\right) \cdot \sum_{t=1}^{T} \mathbb{I}_{\{I_t(h_{1:t-1})=i\}} \\
&\leq T \sum_{\substack{h_{1:T-1} \in \Sigma^{T-1} \\ p_2^*(h_{1:T-1}) \geq p_1^*(h_{1:T-1})}} \left(p_2^*(h_{1:T-1}) - p_1^*(h_{1:T-1})\right) \qquad \text{(A.6)} \\
&= \frac{T}{2} \left\|p_2^* - p_1^*\right\|_1 \\
&\leq T\sqrt{D(p_2^* \parallel p_1^*)/2}\,,
\end{aligned}$$

where the last step is an application of Pinsker's inequality [Cover and Thomas, 2006, Lemma 12.6.1] to distributions p_1^* and p_2^*. Using the chain rule for

[3] It seems from the proof that $N_{\text{rev}}^{(k)}$ could be slightly sharpened to $N_{\text{rev}}^{(k,T-1)} = \sum_{t=1}^{T-1} \mathbb{P}_k\left(I_t \in \mathcal{R}\right)$.

KL divergence [Cover and Thomas, 2006, Theorem 2.5.3] we can write (with somewhat sloppy notation)

$$D(p_2^* \parallel p_1^*) = \sum_{t=1}^{T-1} D\left(p_2^*(h_t \mid h_{1:t-1}) \parallel p_1^*(h_t \mid h_{1:t-1})\right) ,$$

where the t^{th} conditional KL divergence term is

$$\sum_{h_{1:t-1} \in \Sigma^{t-1}} \mathbb{P}_2\left(H_{1:t-1} = h_{1:t-1}\right) \times \sum_{h_t \in \Sigma} \mathbb{P}_2\left(H_t = h_t \mid H_{1:t-1} = h_{1:t-1}\right) \ln \frac{\mathbb{P}_2\left(H_t = h_t \mid H_{1:t-1} = h_{1:t-1}\right)}{\mathbb{P}_1\left(H_t = h_t \mid H_{1:t-1} = h_{1:t-1}\right)} . \tag{A.7}$$

Decompose this sum for the case $I_t(h_{1:t-1}) \notin \mathcal{R}$ and $I_t(h_{1:t-1}) \in \mathcal{R}$. In the first case, we play a none-revealing action, thus our observation $H_t = \mathbf{H}[I_t(h_{1:t-1}), J_t] = \mathbf{H}[I_t(h_{1:t-1}), 1]$ is a deterministic constant in both models 1 and 2, thus both $\mathbb{P}_1\left(\cdot \mid H_{1:t-1} = h_{1:t-1}\right)$ and $\mathbb{P}_2\left(\cdot \mid H_{1:t-1} = h_{1:t-1}\right)$ are degenerate and the KL divergence factor is 0. Otherwise, playing a revealing action, $H_t = \mathbf{H}[I_t(h_{1:t-1}), J_t]$ is the same deterministic function of J_t (which is independent of $H_{1:t-1}$) in both models 1 and 2, and so the inner sum in (A.7) is

$$\sum_{h_t \in \Sigma} \Pr_2[\mathbf{H}[I_t(h_{1:t-1}), J_t] = h_t] \ln \frac{\Pr_2[\mathbf{H}[I_t(h_{1:t-1}), J_t] = h_t]}{\Pr_1[\mathbf{H}[I_t(h_{1:t-1}), J_t] = h_t]} . \tag{A.8}$$

Since $\mathbb{P}_k\left(\mathbf{H}[I_t(h_{1:t-1}), J_t] = h_t\right) = \sum_{j_t \in \underline{M}: \mathbf{H}[I_t(h_{1:t-1}), j_t] = h_t} p_k(j_t)$ $(k = 1{,}2)$, using the log sum inequality [Cover and Thomas, 2006, Theorem 2.7.1]), (A.8) is upper bounded by

$$\sum_{h_t \in \Sigma} \sum_{j_t \in \underline{M}: \mathbf{H}[I_t(h_{1:t-1}), j_t] = h_t} p_2(j_t) \ln \frac{p_2(j_t)}{p_1(j_t)} = \sum_{j_t \in \underline{M}} p_2(j_t) \ln \frac{p_2(j_t)}{p_1(j_t)} = D(p_2 \parallel p_1) .$$

Hence, $D(p_2^* \parallel p_1^*)$ is upper bounded by

$$\sum_{t=1}^{T-1} \sum_{\substack{h_{1:t-1} \in \Sigma^{t-1} \\ I_t(h_{1:t-1}) \in \mathcal{R}}} \mathbb{P}_2\left(H_{1:t-1} = h_{1:t-1}\right) D(p_2 \parallel p_1)$$
$$= D(p_2 \parallel p_1) \sum_{t=1}^{T-1} \Pr_2[I_t \in \mathcal{R}] = D(p_2 \parallel p_1) N_{\text{rev}}^{(2,T-1)} ,$$

where $N_{\text{rev}}^{(k,T-1)} = \sum_{t=1}^{T-1} \mathbb{P}_k\left(I_t \in \mathcal{R}\right)$. This together with (A.6) gives $N_i^{(2)} - N_i^{(1)} \leq T\sqrt{D(p_2 \parallel p_1) N_{\text{rev}}^{(2,T-1)}/2}$.

If $\mathcal{A}$ is random and its internal random "bits" are represented by a random value Z (which is independent of $J_1, J_2, \dots$), then $N_i^{(k)} = \mathbb{E}\left[\tilde{N}_i^{(k)}(Z)\right]$ for $\tilde{N}_i^{(k)}(Z) = \sum_{t=1}^{T} \mathbb{P}_k\left(I_t = i | Z\right)$. Also let $\tilde{N}_{\mathrm{rev}}^{(k,T-1)}(Z) = \sum_{t=1}^{T-1} \mathbb{P}_k\left(I_t \in \mathcal{R} | Z\right)$. The proof above implies that for any fixed $z \in \mathrm{Range}(Z)$,

$$\tilde{N}_i^{(2)}(z) - \tilde{N}_i^{(1)}(z) \leq T\sqrt{D(p_2 \parallel p_1)\tilde{N}_{\mathrm{rev}}^{(2,T-1)}(z)/2}\,,$$

and thus, using also Jensen's inequality,

$$\begin{aligned} N_i^{(2)} - N_i^{(1)} &= \mathbb{E}\left[\tilde{N}_i^{(2)}(Z) - \tilde{N}_i^{(1)}(Z)\right] \\ &\leq \mathbb{E}\left[T\sqrt{D(p_2 \parallel p_1)\tilde{N}_{\mathrm{rev}}^{(2,T-1)}(Z)/2}\right] \\ &\leq T\sqrt{D(p_2 \parallel p_1)\,\mathbb{E}\left[\tilde{N}_{\mathrm{rev}}^{(2,T-1)}(Z)\right]/2} = T\sqrt{D(p_2 \parallel p_1)N_{\mathrm{rev}}^{(2,T-1)}/2}\,, \end{aligned}$$

that is clearly upper bounded by $T\sqrt{D(p_2 \parallel p_1)N_{\mathrm{rev}}^{(2)}/2}$ yielding the statement of the lemma. □

Lemma 9. *Let $\mathbf{G}$ be a non-degenerate game with two outcomes. Let $\mathbf{G}'$ be the game we get by removing the degenerate non-revealing actions from $\mathbf{G}$. Then $\mathrm{R}_T(\mathbf{G}) = \mathrm{R}_T(\mathbf{G}')$.*

Proof. We prove the lemma by showing that for every algorithm $\mathcal{A}$ on game $\mathbf{G}$ there exists an algorithm $\mathcal{A}'$ on $\mathbf{G}'$ such that for any outcome sequence, $\mathrm{R}_T(\mathcal{A}', \mathbf{G}') \leq \mathrm{R}_T(\mathcal{A}, \mathbf{G})$ and vice versa. Recall that the minimax regret of a game is

$$\mathrm{R}_T(\mathbf{G}) = \inf_{\mathcal{A}} \sup_{J_{1:T} \in \underline{M}^T} \mathrm{R}_T(\mathcal{A}, \mathbf{G})\,,$$

where

$$\mathrm{R}_T(\mathcal{A}, \mathbf{G}) = \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{L}[I_t, J_t] - \min_{i \in \underline{N}} \sum_{t=1}^{T} \mathbf{L}[i, J_t]\right]\,.$$

First we observe that the term $\mathbb{E}[\min_{i \in \underline{N}} \sum_{t=1}^{T} \mathbf{L}[i, J_t]]$ does not change by removing degenerate actions. Indeed, by the definition of degenerate action, if the minimum is given by a degenerate action then there exists a non-degenerate action with the same cumulative loss. It follows that we only have to deal with the term $\mathbb{E}[\sum_{t=1}^{T} \mathbf{L}[I_t, J_t]]$.

1. Let $\mathcal{A}'$ be an algorithm on $\mathbf{G}'$. We define the algorithm $\mathcal{A}$ on $\mathbf{G}$ by choosing the same actions as $\mathcal{A}'$ at every time step. Since the action set of $\mathbf{G}$ is a superset of that of $\mathbf{G}'$, this construction results in a well defined algorithm on $\mathbf{G}$, and trivially has the same expected loss as $\mathcal{A}'$.

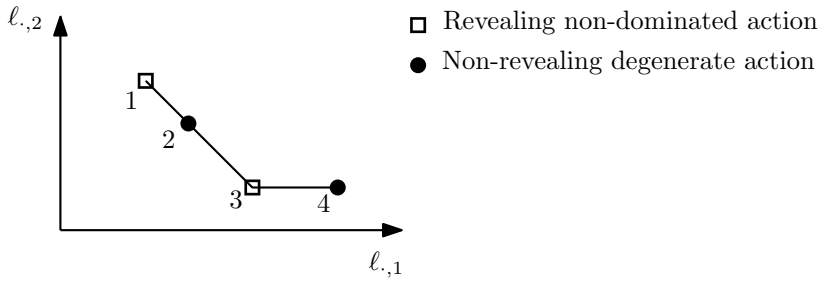


Figure A.1: Degenerate non-revealing actions on the chain. The loss vector of action 2 is a convex combination of that of action 1 and 3. On the other hand, the loss vector of action 4 is component-wise lower bounded by that of action 3.

2. Let $\mathcal{A}$ be an algorithm on $\mathbf{G}$. From the definition of degenerate actions, we know that for every degenerate action i, there are two possibilities:

   (a) There exists a non-degenerate action i_1 such that ℓ_i is component-wise lower bounded by ℓ_{i_1}.

   (b) There are two non-degenerate actions i_1 and i_2 such that ℓ_i is a convex combination of ℓ_{i_1} and ℓ_{i_2}, that is, $\ell_i = \alpha_i \ell_{i_1} + (1-\alpha_i)\ell_{i_2}$ for some $\alpha_i \in (0,1)$.

   An illustration of these cases can be found in Figure A.1. We construct $\mathcal{A}'$ the following way. At every time step t, if $I_t^{\mathcal{A}}$ (the action that algorithm $\mathcal{A}$ would take) is non-degenerate then let $I_t^{\mathcal{A}'} = I_t^{\mathcal{A}}$. If $I_t^{\mathcal{A}} = i$ is a degenerate action of the first kind, let $I_t^{\mathcal{A}'}$ be i_1. If $I_t^{\mathcal{A}} = i$ is a degenerate action of the second kind then let $I_t^{\mathcal{A}'}$ be i_1 with probability α_i and i_2 with probability $1-\alpha_i$. Recall that $\mathbf{G}$ is non-degenerate, so i has to be a non-revealing action. However, i_1 and/or i_2 might be revealing ones. To handle this, $\mathcal{A}'$ is defined to map the observation sequence, before using it as the argument of I_t, replacing the feedbacks corresponding to degenerate action i by $\mathbf{H}[i,1] = \mathbf{H}[i,2]$. That is, intuitively, $\mathcal{A}'$ "pretends" that the feedbacks at such time steps are irrelevant. It is clear that the expected loss of $\mathcal{A}'$ in every time step is less than or equal to the expected loss of $\mathcal{A}$, concluding the proof.

□

Lemma 10. *There exists a constant $c > 0$ (depending on α only) such that for any $\epsilon > 0$,*

$$N_2^{(1)} \geq N_2^{(2)} - cT\epsilon\sqrt{N_{\geq 3}^{(2)}} \qquad \text{and} \qquad N_1^{(2)} \geq N_1^{(1)} - cT\epsilon\sqrt{N_{\geq 3}^{(1)}}\,.$$

Proof. We only prove the first inequality, the other one is symmetric. Using Lemma 8 with $M = 2$, $i = 2$ and the fact that actions 1 and 2 are non-revealing, we have

$$N_2^{(2)} - N_2^{(1)} \leq T\sqrt{D(p_2 \parallel p_1) N_{\geq 3}^{(2)}/2}\,.$$

Lemma 7 with $M = 2$, $p = (\alpha, 1-\alpha)^\top$, and $\epsilon = (\epsilon, -\epsilon)^\top$ gives $D(p_2 \parallel p_1) \leq \hat{c}\epsilon^2$, where $\hat{c}$ depends only on α. Rearranging and substituting $c = \sqrt{\hat{c}/2}$ yields the first statement of the lemma. □

A.2 Lemma from Chapter 5

Lemma 13. *If $\ell \notin \operatorname{Im} \mathbf{A}^\top$ then $\mathbf{G}_1$ is trivial or hopeless.*

Proof. The condition $\ell \notin \operatorname{Im} \mathbf{A}^\top$ implies $\langle \ell \rangle \nsubseteq \operatorname{Im} \mathbf{A}^\top$, that is equivalent to $\ell^\perp \not\supseteq \operatorname{Ker} \mathbf{A}$, which can be seen by taking the orthogonal complement of both sides and using $(\operatorname{Ker} \mathbf{A})^\perp = \operatorname{Im} \mathbf{A}^\top$. The latter implies that there exists v such that $v \in \operatorname{Ker} \mathbf{A}$ but $\ell^\top v \neq 0$. By scaling we can assume w.l.o.g. that $\ell^\top v = 1$. Note that, since the first m_1 rows of $\mathbf{A}$ add up to $\mathbf{1}^\top$ and $v \in \operatorname{Ker} \mathbf{A}$, the coordinates of v sum to zero.

We identify the set of all probability distributions over the set of outcomes $\underline{M}$ with the probability simplex $\Delta_M = \{p \in \mathbb{R}^M \ : \ \sum_{j=1}^M p(j) = 1, \ \forall j \in \underline{M}, \ \ p(j) \geq 0\}$. If $p \in \Delta_M$ is a distribution, then it is easy to see that the first m_1 coordinates of $\mathbf{A}p$ give the probability distribution of observing the different values of the first row of $\mathbf{H}_0$ while the learner chooses action 1 assuming the opponent chooses her actions from p. The same applies to the last m_2 coordinates of $\mathbf{A}p$ and action 2. It follows that if $\mathbf{A}p_1 = \mathbf{A}p_2$ for two distributions then no algorithm can distinguish them. We find such p_1, p_2 and apply this idea as follows:

If for all $p \in \Delta_M$, $\ell^\top p \geq 0$ (or $\ell^\top p \leq 0$), then $\mathbf{G}_1$ has zero minimax regret and thus it is trivial. Otherwise, there exist p_+ and p_- in Δ_M with $\ell^\top p_+ > 0$ and $\ell^\top p_- < 0$. Now either there exists $p_0 \in \operatorname{Int}(\Delta_M)$ such that $\ell^\top p_0 = 0$, or we can assume w.l.o.g. that one of p_+ and p_- is in $\operatorname{Int}(\Delta_M)$, in which case there must be again a $p_0 \in \operatorname{Int}(\Delta_M)$ on the segment $\overline{p_+ p_-}$ such that $\ell^\top p_0 = 0$ by the continuity of $\ell^\top p$ in p. In other words, we have a distribution p_0 over $\underline{M}$ such that p_0 is not on the boundary of Δ_M and the expected loss of the two actions are equal.

Now let $p_1 = p_0 + \varepsilon v$ and $p_2 = p_0 - \varepsilon v$ for some $\varepsilon > 0$. If ε is small enough then both p_1 and p_2 are in Δ_M. Since $\mathbf{A}v = 0$ we have that $\mathbf{A}p_1 = \mathbf{A}p_0 = \mathbf{A}p_2$. On the other hand, $\ell^\top p_1 = \varepsilon > 0$ and $\ell^\top p_2 = -\varepsilon < 0$ imply that action k is optimal under p_k for $k = 1,2$.

Fix any strategy $\mathcal{A}$ of the learner. We use randomization replacing the outcomes by a sequence $J_1, J_2, \ldots, J_T \in \underline{M}^T$ of random variables i.i.d. according to p_k, $k \in \{1, 2\}$, and independently of the internal randomization of $\mathcal{A}$. Let

$$N_i^{(k)} = N_i^{(k)}(\mathcal{A}, T) \stackrel{\triangle}{=} \sum_{t=1}^{T} \mathbb{P}_k \left(I_t = i \right) \in [0, T]$$

be the expected number of times action i is chosen by $\mathcal{A}$ under p_k up to time step T. With subindex k, $\mathbb{P}_k(\cdot)$ and $\mathbb{E}_k[\cdot]$ denote probability and expectation

given outcome model $k \in \{1,2\}$, respectively. Then, the worst case regret of $\mathcal{A}$ is

$$\mathrm{R}_T^{\mathcal{A}*}(\mathbf{G}_1) \geq N_{3-k}^{(k)}(\ell_{3-k}-\ell_k)^\top p_k = N_{3-k}^{(k)}\varepsilon = \begin{cases} N_2^{(1)}\varepsilon & \text{if } k=1, \\ (T-N_2^{(2)})\varepsilon & \text{if } k=2, \end{cases}$$

due to $\ell_1 = 0$, $\ell_2 = \ell$, $\ell^\top p_1 = -\ell^\top p_2 = \varepsilon$, and $N_1^{(2)} + N_2^{(2)} = T$. Observe that $\mathbf{A}p_1 = \mathbf{A}p_2$ means that for both actions, the feedback distribution is the same under outcome distributions p_1 and p_2, implying (by induction) that for each $t \geq 1$, $\mathbb{P}_1(I_t = 2) = \mathbb{P}_2(I_t = 2)$. This leads to $N_2^{(1)} = N_2^{(2)} \triangleq N_2 = N_2(\mathcal{A}, T)$. Thus, we have

$$\mathrm{R}_T(\mathbf{G}_1) = \inf_{\mathcal{A}} \mathrm{R}_T^{\mathcal{A}}(\mathbf{G}_1) \geq \inf_{\mathcal{A}} \max_{k\in\underline{2}} N_{3-k}^{(k)}\varepsilon = \varepsilon \inf_{\mathcal{A}} \max(N_2, T-N_2) \geq \varepsilon T/2,$$

that is, $\mathbf{G}_1$ is hopeless. □

A.3 Lemmas from Chapter 6

Lemma 14. *For any $n \geq 1$ and i, j such that $C_i, C_j \in \mathcal{C}$, $\mathbb{E}_{n-1}[\hat{\delta}_{i,j}(n)] = \delta_{i,j}$.*

Proof. Consider first the case when actions i and j are neighbors. In this case,

$$\hat{\delta}_{i,j}(n) = \sum_{k\in N_{i,j}^+} Y_k(n)^\top v_{i,j,k} = \sum_{k\in N_{i,j}^+} (S_k u_k(n))^\top v_{i,j,k} = \sum_{k\in N_{i,j}^+} u_k(n)^\top S_k^\top v_{i,j,k}\,,$$

and thus

$$\begin{aligned}\mathbb{E}_{n-1}\left[\hat{\delta}_{i,j}(n)\right] &= \sum_{k\in N_{i,j}^+} \mathbb{E}_{n-1}\left[u_k(n)^\top\right] S_k^\top v_{i,j,k} = p^{*\top} \sum_{k\in N_{i,j}^+} S_k^\top v_{i,j,k} \\ &= p^{*\top}(\ell_i - \ell_j) = \delta_{i,j}\,.\end{aligned}$$

For non-adjacent i and j, we have a telescoping sum:

$$\begin{aligned}\mathbb{E}_{n-1}\left[\hat{\delta}_{i,j}(n)\right] &= \sum_{k=1}^{r} \mathbb{E}_{n-1}[\hat{\delta}_{i_{k-1},i_k}(n)] \\ &= p^{*\top}\left(\ell_{i_0} - \ell_{i_1} + \ell_{i_1} - \ell_{i_2} + \cdots + \ell_{i_{r-1}} - \ell_{i_r}\right) = \delta_{i,j}\,,\end{aligned}$$

where $i = i_0, i_1, \ldots, i_r = j$ is the path the algorithm uses in round n, known at the end of round $n-1$. □

Lemma 15. *The conditional variance of $\hat{\delta}_{i,j}(n)$, $\mathbb{V}\mathrm{ar}_{n-1}(\hat{\delta}_{i,j}(n))$, is upper bounded by $V = 2\sum_{\{i,j\in\mathcal{L}\}} \sum_{k\in N_{i,j}^+} \|v_{i,j,k}\|_2^2$. The range of the estimates $\hat{\delta}_{i,j}(n)$ is upper bounded by $R = \sum_{\{i,j\}\in\mathcal{L}} \sum_{k\in N_{i,j}^+} \|v_{i,j,k}\|_1$.*

Proof. For neighboring cells i, j, we write

$$\begin{aligned}
\hat{\delta}_{i,j}(n) &= \sum_{k \in N^+_{i,j}} Y_k(n)^\top v_{i,j,k} \qquad \text{and thus} \\
&\mathbb{V}\mathrm{ar}_{n-1}(\hat{\delta}_{i,j}(n)) \\
&= \mathbb{V}\mathrm{ar}_{n-1}\left(\sum_{k \in N^+_{i,j}} Y_k(n)^\top v_{i,j,k}\right) \\
&= \sum_{k \in N^+_{i,j}} \mathbb{E}_{n-1}\left[v_{i,j,k}^\top (Y_k(n) - \mathbb{E}_{n-1}[Y_k(n)])(Y_k(n) - \mathbb{E}_{n-1}[Y_k(n)])^\top v_{i,j,k}\right] \\
&\leq \sum_{k \in N^+_{i,j}} \|v_{i,j,k}\|_2^2 \;\; \mathbb{E}_{n-1}\left[\|Y_k(n) - \mathbb{E}_{n-1}[Y_k(n)]\|_2^2\right] \\
&\leq \sum_{k \in N^+_{i,j}} \|v_{i,j,k}\|_2^2 \,, \qquad \text{(A.9)}
\end{aligned}$$

where in (A.9) we used that $Y_k(n)$ is a unit vector and $\mathbb{E}_{n-1}[Y_k(n)]$ is a probability vector.

For i, j non-neighboring cells, let $i = i_0, i_1, \ldots, i_r = j$ the path used for the estimate in round n. Then $\hat{\delta}_{i,j}(n)$ can be written as

$$\hat{\delta}_{i,j}(n) = \sum_{s=1}^{r} \hat{\delta}_{i_{s-1},i_s}(n) = \sum_{s=1}^{r} \sum_{k \in N^+_{i_{s-1},i_s}} Y_k(n)^\top v_{i_{s-1},i_s,k} \,.$$

It is not hard to see that an action can only be in at most two neighborhood action sets in the path and so the double sum can be rearranged as

$$\sum_{k \in \bigcup N^+_{i_{s-1},i_s}} Y_k(n)^\top (v_{i_{s_k-1},i_{s_k},k} + v_{i_{s_k} i_{s_k+1},k}) \,,$$

and thus

$$\mathbb{V}\mathrm{ar}_{n-1}\left(\hat{\delta}_{i,j}(n)\right) \leq 2 \sum_{s=1}^{r} \sum_{k \in N^+_{s-1,s}} \|v_{i_{s-1},i_s,k}\|_2^2 \leq V \,.$$

The bound of the range trivially follows from the definition of the estimates. □

Lemma 16. *Let action i be a degenerate action. Let $N_i^+ = \{j \;:\; C_j \in \mathcal{C}, C_i \subset C_j\}$. The following two statements hold:*

1. *If any of the actions in N_i^+ is eliminated, then action i is eliminated as well.*

2. *There exists an action $k_i \in N_i^+$ such that $\delta_{k_i,j^*} \geq \delta_{i,j^*}$.*

Proof. 1. In an elimination set, we eliminate every action whose cell is contained in a closed half space. Let us assume that $j \in N_i^+$ is being eliminated. According to the definition of N_i^+, $C_i \subset C_j$ and thus C_i is also contained in the half space.

2. First let us assume that p^* is not in the affine subspace spanned by C_i. Let p be an arbitrary point in the relative interior of C_i. We define the point $p' = p + \varepsilon(p - p^*)$. For a small enough $\varepsilon > 0$, $p' \in C_k \in N_i^+$, and at the same time, $p' \notin C_i$. Thus we have

$$\begin{aligned}
\ell_k^\top \left(p + \varepsilon \left(p - p^*\right)\right) &\leq \ell_i^\top \left(p + \varepsilon \left(p - p^*\right)\right) \\
(1+\varepsilon)\ell_k^\top p - \varepsilon \ell_k^\top p^* &\leq (1+\varepsilon)\ell_i^\top p - \varepsilon \ell_i^\top p^* \\
-\varepsilon \ell_k^\top p^* &\leq -\varepsilon \ell_i^\top p^* \\
\ell_k^\top p^* &\geq \ell_i^\top p^* \\
\delta_{k,j^*} &\geq \delta_{i,j^*} \,,
\end{aligned}$$

where we used that $\ell_k^\top p = \ell_i^\top p$.

For the case when p^* lies in the affine subspace spanned by C_i, We take a hyperplane that contains the affine subspace. Then we take an infinite sequence $(p_n)_n$ such that every element of the sequence is in the same side of the hyperplane, $p_n \neq p^*$ and the sequence converges to p^*. Then the statement is true for every element p_n and, since the value $\delta_{r,s}$ is continuous in p, the limit has the desired property as well.

□

Lemma 17. *There exists a (problem dependent) constant c such that for any small enough ε, the following inequalities hold:*

$$\begin{aligned}
N_1^2 &\geq N_1^1 - cT\varepsilon\sqrt{N_4^1}\,, & N_3^2 &\geq N_3^1 - cT\varepsilon\sqrt{N_4^1}\,, \\
N_2^1 &\geq N_2^2 - cT\varepsilon\sqrt{N_4^2}\,, & N_3^1 &\geq N_3^2 - cT\varepsilon\sqrt{N_4^2}\,.
\end{aligned}$$

Proof. For any $1 \leq t \leq T$, let $f^t = (f_1, \ldots, f_t) \in \Sigma^t$ be a feedback sequence up to time step t. For $i = 1, 2$, let p_i^* be the probability mass function of feedback sequences of length $T - 1$ under opponent strategy p_i and algorithm $\mathcal{A}$. We start by upper bounding the difference between values under the two

opponent strategies. For $i \neq j \in \{1,2\}$ and $k \in \{1,2,3\}$,

$$\begin{aligned} N_k^i - N_k^j &= \sum_{f^{T-1}} \big(p_i^*(f^{T-1}) - p_j^*(f^{T-1})\big) \sum_{t=0}^{T-1} \mathbb{I}_{\{\mathcal{A}(f^t)\in A_k\}} \\ &\leq \sum_{\substack{f^{T-1}:\\ p_i^*(f^{T-1})-p_j^*(f^{T-1})\geq 0}} \big(p_i^*(f^{T-1}) - p_j^*(f^{T-1})\big) \sum_{t=0}^{T-1} \mathbb{I}_{\{\mathcal{A}(f^t)\in A_k\}} \\ &\leq T \sum_{\substack{f^{T-1}:\\ p_i^*(f^{T-1})-p_j^*(f^{T-1})\geq 0}} p_i^*(f^{T-1}) - p_j^*(f^{T-1}) = \frac{T}{2}\|p_1^* - p_2^*\|_1 \\ &\leq T\sqrt{\mathrm{KL}(p_1^*||p_2^*)/2}\,, \end{aligned} \tag{A.10}$$

where $\mathrm{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence and $\|\cdot\|_1$ is the L_1-norm. The last inequality follows from Pinsker's inequality [Cover and Thomas, 2006]. To upper bound $\mathrm{KL}(p_1^*||p_2^*)$ we use the chain rule for KL-divergence. By overloading p_i^* so that $p_i^*(f^{t-1})$ denotes the probability of feedback sequence f^{t-1} under opponent strategy p_i and algorithm $\mathcal{A}$, and $p_i^*(f_t|f^{t-1})$ denotes the conditional probability of feedback $f_t \in \Sigma$ given that the past feedback sequence was f^{t-1}, again under p_i and $\mathcal{A}$. With this notation we have

$$\begin{aligned} \mathrm{KL}(p_1^*||p_2^*) &= \sum_{t=1}^{T-1} \sum_{f^{t-1}} p_1^*(f^{t-1}) \sum_{f_t} p_1^*(f_t|f^{t-1}) \log \frac{p_1^*(f_t|f^{t-1})}{p_2^*(f_t|f^{t-1})} \\ &= \sum_{t=1}^{T-1} \sum_{f^{t-1}} p_1^*(f^{t-1}) \sum_{i=1}^{4} \mathbb{I}_{\{\mathcal{A}(f^{t-1})\in A_i\}} \sum_{f_t} p_1^*(f_t|f^{t-1}) \log \frac{p_1^*(f_t|f^{t-1})}{p_2^*(f_t|f^{t-1})} \end{aligned} \tag{A.11}$$

Let $a_{f_t}^\top$ be the row of S that corresponds to the feedback symbol f_t.[4] Assume $k = \mathcal{A}(f^{t-1})$. If the feedback set of action k does not contain f_t then trivially $p_i^*(f_t|f^{t-1}) = 0$ for $i = 1, 2$. Otherwise $p_i^*(f_t|f^{t-1}) = a_{f_t}^\top p_i$. Since $p_1 - p_2 = 2\varepsilon v$ and $v \in \operatorname{Ker} S$, we have $a_{f_t}^\top v = 0$ and thus, if the choice of the algorithm is in either A_1, A_2 or A_3, then $p_1^*(f_t|f^{t-1}) = p_2^*(f_t|f^{t-1})$. It follows that the inequality chain can be continued from (A.11) by writing

$$\begin{aligned} \mathrm{KL}(p_1^*||p_2^*) &\leq \sum_{t=1}^{T-1} \sum_{f^{t-1}} p_1^*(f^{t-1}) \mathbb{I}_{\{\mathcal{A}(f^{t-1})\in A_4\}} \sum_{f_t} p_1^*(f_t|f^{t-1}) \log \frac{p_1^*(f_t|f^{t-1})}{p_2^*(f_t|f^{t-1})} \\ &\leq c_1\varepsilon^2 \sum_{t=1}^{T-1} \sum_{f^{t-1}} p_1^*(f^{t-1}) \mathbb{I}_{\{\mathcal{A}(f^{t-1})\in A_4\}} \\ &\leq c_1\varepsilon^2 N_4^1\,. \end{aligned} \tag{A.12}$$

[4] Recall that we assumed that different actions have difference feedback symbols, and thus a row of S corresponding to a symbol is unique.

In (A.12) we used Lemma 6 from Section 4.5 to upper bound the KL-divergence of p_1 and p_2. Flipping p_1^* and p_2^* in (A.10) we get the same result with N_4^2. Reading together with the bound in (A.10) we get all the desired inequalities. □

The following lemma concerns the problem of producing an estimate of an unknown mean of some stochastic process with a given relative error bound and with high probability in a sample-efficient manner. The procedure is a simple variation of the one proposed by Mnih et al. [2008]. The main differences are that here we deal with martingale difference sequences shifted by an unknown constant, which becomes the common mean, whereas Mnih et al. [2008] considered an i.i.d. sequence. On the other hand, we consider the case when we have a known upper bound on the predictable variance of the process, whereas one of the main contributions of Mnih et al. [2008] was the lifting of this assumption. The proof of the lemma is omitted, as it follows the same lines as the proof of results of Mnih et al. [2008] (the details of these proofs are found in the thesis of [Mnih, 2008]), the only difference being, that here we would need to use Bernstein's inequality for martingales, in place of the empirical Bernstein inequality, which was used by Mnih et al. [2008].

Lemma 28. *Let $(\mathcal{F}_t)$ be a filtration on some probability space, and let (X_t) be an $\mathcal{F}_t$-adapted sequence of random variables. Assume that (X_t) is such that, almost surely, the range of each random variable X_t is bounded by $R > 0$, $\mathbb{E}_{X_t|\mathcal{F}_{t-1}}[=]\mu$, and $\mathbb{V}ar[X_t|\mathcal{F}_{t-1}] \le \sigma^2$ a.s., where R, $\mu \neq 0$ and σ^2 are non-random constants. Let $p > 1$, $\epsilon > 0$, $0 < \delta < 1$ and let*

$$L_n = (1+\varepsilon) \max_{1\le t\le n} \left\{ |\overline{X}_t| - c_t \right\}, \quad \text{and} \quad U_n = (1-\varepsilon) \min_{1\le t\le n} \left\{ |\overline{X}_t| + c_t \right\},$$

where $c_t = c(\sigma, R, t, \delta)$, and $c(\cdot)$ is defined in (6.1). Define the estimate $\hat{\mu}_n$ of μ as follows:

$$\hat{\mu}_n = \operatorname{sgn}(\overline{X}_n) \, \frac{(1+\varepsilon)L_n + (1-\varepsilon)U_n}{2}.$$

Denote the stopping time $\tau = \min\{n \ : \ L_n \ge U_n\}$. Then, with probability at least $1-\delta$,

$$|\hat{\mu}_\tau - \mu| \le \varepsilon\,|\mu| \qquad \text{and} \qquad \tau \le C \cdot \max\left(\frac{\sigma^2}{\epsilon^2\mu^2}, \frac{R}{\epsilon|\mu|}\right)\left(\log\frac{1}{\delta} + \log\frac{R}{\epsilon|\mu|}\right),$$

where $C > 0$ is a universal constant.

A.4 Lemmas from Chapter 7

Lemma 18. *For any $\{i,j\} \in \mathcal{N}$, $t \ge 1$,*

$$\mathbb{P}\left(|\tilde{\delta}_{i,j}(t) - \delta_{i,j}| \ge c_{i,j}(t)\right) \le 2|N_{i,j}^+|t^{1-2\alpha}.$$

Proof.

$$
\begin{aligned}
&\mathbb{P}\left(|\tilde{\delta}_{i,j}(t) - \delta_{i,j}| \geq c_{i,j}(t)\right) \\
&\leq \sum_{k \in N_{i,j}^+} \mathbb{P}\left(|v_{i,j,k}^\top \frac{\nu_k(t-1)}{n_k(t-1)} - v_{i,j,k}^\top S_k p^*| \geq \|v_{i,j,k}\|_\infty \sqrt{\frac{\alpha \log t}{n_k(t-1)}}\right) && \text{(A.13)} \\
&= \sum_{k \in N_{i,j}^+} \sum_{s=1}^{t-1} \mathbb{I}_{\{n_k(t-1)=s\}} \mathbb{P}\left(|v_{i,j,k}^\top \frac{\nu_k(t-1)}{s} - v_{i,j,k}^\top S_k p^*| \geq \|v_{i,j,k}\|_\infty \sqrt{\frac{\alpha \log t}{s}}\right) && \text{(A.14)} \\
&\leq \sum_{k \in N_{i,j}^+} 2t^{1-2\alpha} && \text{(A.15)} \\
&= 2|N_{i,j}^+| t^{1-2\alpha}\,,
\end{aligned}
$$

where in (A.13) we used the triangle inequality and the union bound and in (A.15) we used Hoeffding's inequality. □

Lemma 19. *Take an action i and a plausible pair $(\mathcal{P}', \mathcal{N}') \in \Psi$ such that $i \in \mathcal{P}'$. Then there exists a path π that starts at i and ends at i^* that lies in $\mathcal{N}'$.*

Proof. If $(\mathcal{P}', \mathcal{N}')$ is a valid configuration, then there is a convex polytope $\Pi \subseteq \Delta_M$ such that $p^* \in \Pi$, $\mathcal{P}' = \{i \ : \ \dim \mathcal{C}_i \cap \Pi = M - 1\}$ and $\mathcal{N}' = \{\{i, j\} \ : \ \dim \mathcal{C}_i \cap \mathcal{C}_j \cap \Pi = M - 2\}$.

Let p' be an arbitrary point in $\mathcal{C}_i \cap \Pi$. We enumerate the actions whose cells intersect with the line segment $\overline{p'p^*}$, in the order as they appear on the line segment. We show that this sequence of actions $i_0, \ldots, i_r$ is a feasible path.

- It trivially holds that $i_0 = i$, and i_r is optimal.
- It is also obvious that consecutive actions on the sequence are in $\mathcal{N}'$.

For an illustration we refer the reader to Figure A.2 □

Lemma 20. *Take any action k. On the event $\mathcal{E}^c$, from $I_t = k$ it follows that*

$$
n_k(t-1) \leq 4W_k^2 \frac{d_k^2}{\delta_k^2} \alpha \log t\,.
$$

Proof. We define the "parent action" k' of k as follows: If k is not degenerate then $k' = k$. If k is degenerate then we define k' to be the Pareto-optimal action such that $\delta_{k'} \geq \delta_k$ and k is in the neighborhood action set of k' and some other Pareto-optimal action. It follows from Lemma 29, stated and proved after the current lemma, that k' is well-defined. We also know that k' must be in $\mathcal{P}(t)$.

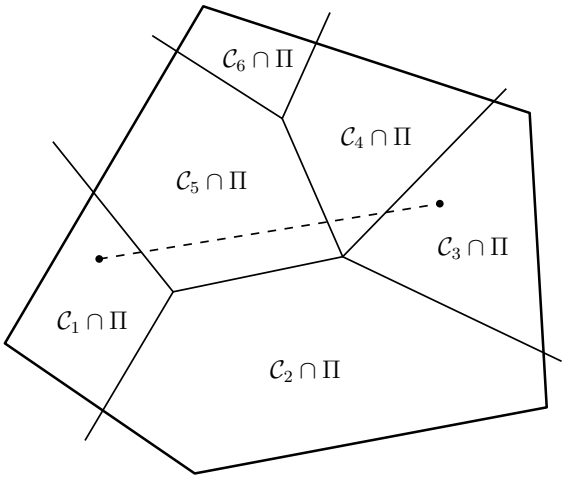


Figure A.2: The dashed line defines the feasible path $1, 5, 4, 3$.

Now, according to Lemma 19, there exists a path $k' = k_0, k_1, \ldots, k_r = i*$. We write

$$\begin{aligned}
\delta_k \leq \delta_{k'} &= \sum_{s=1}^{r} \delta_{s-1,s} \\
&\leq 2 \sum_{s=1}^{r} c_{i_{s-1},i_s} \\
&= 2 \sum_{s=1}^{r} \sum_{j \in N^+_{i_{s-1},i_s}} \|v_{i_{s-1},i_s,j}\|_\infty \sqrt{\frac{\alpha \log t}{n_j(t-1)}} \\
&\leq 2 \sum_{s=1}^{r} \sum_{j \in N^+_{i_{s-1},i_s}} W_k \sqrt{\frac{\alpha \log t}{n_k(t-1)}} \\
&\leq 2 d_k W_k \sqrt{\frac{\alpha \log t}{n_k(t-1)}} \,.
\end{aligned}$$

Rearranging the last inequality yields the statement of the lemma. □

Lemma 29. *Let action i be a degenerate action in the neighborhood action set $N^+_{k,l}$ of neighboring actions k and l. Then ℓ_i is a convex combination of ℓ_k and ℓ_l.*

Proof. For simplicity, we rename the degenerate action i to action 1, while the other actions k, l will be called actions 2 and 3, respectively. Since action 1 is a degenerate action between actions 2 an 3, we have that

$$(p \in \Delta_M \text{ and } p \perp (\ell_1 - \ell_2)) \Rightarrow (p \perp (\ell_1 - \ell_3) \text{ and } p \perp (\ell_2 - \ell_3))$$

implying

$$(\ell_1 - \ell_2)^\perp \subseteq (\ell_1 - \ell_3)^\perp \cap (\ell_2 - \ell_3)^\perp \ .$$

Using de Morgan's law we get

$$\langle \ell_1 - \ell_2 \rangle \supseteq \langle \ell_1 - \ell_3 \rangle \oplus \langle \ell_2 - \ell_3 \rangle \, .$$

This implies that for any $c_1, c_2 \in \mathbb{R}$ there exists a $c_3 \in \mathbb{R}$ such that

$$\begin{aligned} c_3(\ell_1 - \ell_2) &= c_1(\ell_1 - \ell_3) + c_2(\ell_2 - \ell_3) \\ \ell_3 &= \frac{c_1 - c_3}{c_1 + c_2}\ell_1 + \frac{c_2 + c_3}{c_1 + c_2}\ell_2 \, , \end{aligned}$$

suggesting that ℓ_3 is an affine combination of (or collinear with) ℓ_1 and ℓ_2.

We know that there exists $p_1 \in \Delta$ such that $\ell_1^\top p_1 < \ell_2^\top p_1$ and $\ell_1^\top p_1 < \ell_3^\top p_1$. Also, there exists $p_2 \in \Delta_M$ such that $\ell_2^\top p_2 < \ell_1^\top p_2$ and $\ell_2^\top p_2 < \ell_3^\top p_2$. Using these and linearity of the dot product we get that ℓ_3 must be the middle point on the line, which means that ℓ_3 is indeed a convex combination of ℓ_1 and ℓ_2. □

Lemma 21. *For any $\{i, j\} \in \mathcal{N}$, $t \geq 1$,*

$$\mathbb{P}\left(|\tilde{\delta}_{i,j}(t) - \delta_{i,j}| \geq c_{i,j}(t)\right) \leq 2|V_{i,j}|t^{1-2\alpha} \, .$$

Proof. The prof of this lemma is identical to that of Lemma 18 with the difference that we replace all appearance of the neighborhood action sets $N^+_{i,j}$ with the corresponding observer sets $V_{i,j}$. □

Lemma 22. *Fix any $t \geq 1$.*

1. *Take any action i. On the event $\mathcal{E}_t^c \cap \mathcal{D}_t$,[5] from $i \in \mathcal{P}(t) \cup N^+(t)$ it follows that*

$$\delta_i \leq 2d_i\sqrt{\frac{\alpha \log t}{f(t)}} \max_{k \in \underline{N}} \frac{W_k}{\sqrt{\eta_k}} \, .$$

2. *Take any action k. On the event $\mathcal{E}_t^c \cap \mathcal{D}_t^c$, from $I_t = k$ it follows that*

$$n_k(t-1) \leq \min_{j \in \mathcal{P}(t) \cup N^+(t)} 4W_k^2 \frac{d_j^2}{\delta_j^2} \, \alpha \log t \, .$$

[5]Here and in what follows all statements that start with "On event X" should be understood to hold almost surely on the event. However, to minimize clutter we will not add the qualifier "almost surely".

Proof. First we observe that for any neighboring action pair $\{i,j\} \in \mathcal{N}(t)$, on $\mathcal{E}_t^c$ it holds that $\delta_{i,j} \leq 2c_{i,j}(t)$. Indeed, from $\{i,j\} \in \mathcal{N}(t)$ it follows that $\tilde{\delta}_{i,j}(t) \leq c_{i,j}(t)$. Now, on $\mathcal{E}_t^c$, $\delta_{i,j} \leq \tilde{\delta}_{i,j}(t) + c_{i,j}(t)$. Putting together the two inequalities we get $\delta_{i,j} \leq 2c_{i,j}(t)$.

Now, fix some action i that is not dominated. We define the "parent action" i' of i as follows: If i is not degenerate then $i' = i$. If i is degenerate then we define i' to be the Pareto-optimal action such that $\delta_{i'} \geq \delta_i$ and i is in the neighborhood action set of i' and some other Pareto-optimal action. It follows from Lemma 29 that i' is well-defined.

Consider case 1. Thus, $I_t \neq k(t) = \operatorname{argmax}_{j \in \mathcal{P}(t) \cup \mathcal{V}(t)} W_j^2 / n_j(t-1)$. Therefore, $k(t) \notin \mathcal{R}(t)$, *i.e.*, , $n_{k(t)}(t-1) > \eta_{k(t)} f(t)$. Assume now that $i \in \mathcal{P}(t) \cup N^+(t)$. If i is degenerate then i' as defined in the previous paragraph is in $\mathcal{P}(t)$ (because the rejected regions in the algorithm are closed). In any case, by Lemma 19, there is a path $(i_0, \ldots, i_r)$ in $\mathcal{N}(t)$ that connects i' to i^* ($i^* \in \mathcal{P}(t)$ holds on $\mathcal{E}_t^c$). We have that

$$\begin{aligned}
\delta_i \leq \delta_{i'} &= \sum_{s=1}^{r} \delta_{i_{s-1},i_s} \\
&\leq 2 \sum_{s=1}^{r} c_{i_{s-1},i_s} \\
&= 2 \sum_{s=1}^{r} \sum_{j \in V_{i_{s-1},i_s}} \|v_{i_{s-1},i_s,j}\|_\infty \sqrt{\frac{\alpha \log t}{n_j(t-1)}} \\
&\leq 2 \sum_{s=1}^{r} \sum_{j \in V_{i_{s-1},i_s}} W_j \sqrt{\frac{\alpha \log t}{n_j(t-1)}} \\
&\leq 2 d_i W_{k(t)} \sqrt{\frac{\alpha \log t}{n_{k(t)}(t-1)}} \\
&\leq 2 d_i W_{k(t)} \sqrt{\frac{\alpha \log t}{\eta_{k(t)} f(t)}} \,.
\end{aligned}$$

Upper bounding $W_{k(t)}/\sqrt{\eta_{k(t)}}$ by $\max_{k \in \underline{N}} W_k/\sqrt{\eta_k}$ we obtain the desired bound.

Now, for case 2 take an action k, consider $\mathcal{E}^c \cap \mathcal{D}_t^c$, and assume that $I_t = k$. On D_t^c, $I_t = k(t)$. Thus, from $I_t = k$ it follows that $W_k/\sqrt{n_k(t-1)} \geq W_j/\sqrt{n_j(t-1)}$ holds for all $j \in \mathcal{P}(t)$. Let $J_t = \operatorname{argmin}_{j \in \mathcal{P}(t) \cup N^+(t)} \frac{d_j^2}{\delta_j^2}$. Now, similarly to the previous case, there exists a path $(i_0, \ldots, i_r)$ from the parent

action $J'_t \in \mathcal{P}(t)$ of J_t to i^* in $\mathcal{N}(t)$. Hence,

$$\begin{aligned}\delta_{J_t} \le \delta_{J'_t} &= \sum_{s=1}^{r} \delta_{i_{s-1},s} \\ &\le 2\sum_{s=1}^{r} \sum_{j \in V_{i_{s-1},i_s}} W_j \sqrt{\frac{\alpha \log t}{n_j(t-1)}} \\ &\le 2 d_{J_t} W_k \sqrt{\frac{\alpha \log t}{n_k(t-1)}}\,,\end{aligned}$$

implying

$$\begin{aligned}n_k(t-1) &\le 4W_k^2 \frac{d_{J_t}^2}{\delta_{J_t}^2} \alpha \log t \\ &= \min_{j \in \mathcal{P}(t) \cup N^+(t)} 4W_k^2 \frac{d_j^2}{\delta_j^2}\, \alpha \log t\,.\end{aligned}$$

This concludes the proof of Lemma 22. □

Lemma 23. *Let $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ be a finite partial-monitoring game and $p \in \Delta_M$ an opponent strategy. There exists a $\rho_2 > 0$ such that A_{ρ_2} is a point-local game in $\mathbf{G}$.*

Proof. For any (not necessarily neighboring) pair of actions $\{i, j\}$, the boundary between them is defined by the set $B_{i,j} = \{p \in \Delta_M \;:\; (\ell_i - \ell_j)^\top p = 0\}$. We generalize this notion by introducing the *margin*: for any $\xi \ge 0$, let the margin be the set $B_{i,j}^{\xi} = \{p \in \Delta_M \;:\; |(\ell_i - \ell_j)^\top p| \le \xi\}$. It follows from finiteness of the action set that there exists a $\xi^* > 0$ such that for any set K of neighboring action pairs,

$$\bigcap_{\{i,j\} \in K} B_{i,j} \neq \emptyset \qquad \Longleftrightarrow \qquad \bigcap_{\{i,j\} \in K} B_{i,j}^{\xi^*} \neq \emptyset\,. \tag{A.16}$$

Let $\rho_2 = \xi^*/2$. Let $A = A_{\rho_2}$. Then for every pair i, j in A, $(\ell_i - \ell_j)^\top p^* = \delta_{i,j} \le \delta_i + \delta_j \le \rho_2$. That is, $p^* \in B_{i,j}^{\xi^*}$. It follows that $p^* \in \bigcap_{i,j \in A \times A} B_{i,j}^{\xi^*}$. This, together with (A.16), implies that A is a point-local game. □

A.5 Lemmas from Chapter 8

Lemma 24. *Given any expert (s, w) and label y, the* approximation error*, defined as $\min_{(s,w') \in \mathcal{D}_\alpha} |\hat{\ell}(w^\top(s \odot x), y) - \hat{\ell}(w'^\top(s \odot x), y)|$ is upper bounded by $\frac{L\sqrt{d}}{\alpha - 1}$, where L is the Lipschitz constant of $\hat{\ell}$.*

Proof. For any w_1 and w_2,

$$|\hat{\ell}(w_1^\top(s \odot x), y) - \hat{\ell}(w_2^\top(s \odot x), y)| \leq \|w_1 - w_2\| L \,.$$

It follows from the discretization that for any w there exists a point w' in the discretization such that $\|w - w'\| \leq \sqrt{d}/(\alpha - 1)$. This implies the statement of the lemma. □

Lemma 25. *Let e_k denote the k^{th} basis vector of dimension d. Against opponent strategy k, the instantaneous expected regret for any action such that $(s, s_\ell) \neq (e_k, 0)$ is at least $\frac{d\epsilon}{2}$.*

Proof. First, we calculate the expected loss of the action $(w, s, s^L) = (e_k, e_k, 0)$, assuming opponent strategy k:

$$\mathbb{E}_k[\ell_t(e_k, e_k, 0)] = c_k + \frac{1}{2} \underbrace{(1 - a_k - (d-1)\epsilon)}_{\mathbb{P}_k(Z \neq k)} \,.$$

Next, we lower bound the expected loss of any other action. For action $(e_j, e_j, 0)$ with $j \neq k$ we have

$$\mathbb{E}_k[\ell_t(e_j, e_j, 0)] = c_j + \frac{1}{2} \underbrace{(1 - a_j + \epsilon)}_{\mathbb{P}_k(Z \neq j)} \,.$$

It is clear that requesting more than one feature and/or using different w values will lead to greater expected loss. Thus the expected instantaneous regret of any action other than $(e_k, e_k, 0)$ can be lower bounded by the value

$$\begin{aligned}
\mathbb{E}_k[\ell_t(e_j, e_j, 0) - \ell_t(e_k, e_k, 0)] &= c_j - c_k + \frac{1}{2}(1 - a_j + \epsilon) - \frac{1}{2}(1 - a_k - (d-1)\epsilon) \\
&= c_j - c_k - \frac{1}{2}(a_j - a_k) + \frac{d\epsilon}{2} \\
&= \frac{d\epsilon}{2} \,.
\end{aligned}$$

□

Lemma 26. *There exists a constant C_1 such that for any $i, j \in \{1, \ldots, d\}$,*

$$\mathbb{E}_i[N_i] - \mathbb{E}_j[N_i] \leq C_1 T \epsilon \sqrt{d \mathbb{E}_j[N_L]} \,.$$

Proof. For the values x_t, h_t, etc., we denote randomness by capitalization. We denote a sequence of observations $(h_1, \ldots, h_t)$ as h^t. Furthermore, let $\mathcal{A}_t^s(h^{t-1})$ denote the "s-component" of the action taken by the algorithm at time step t.

Similarly, $\mathcal{A}_t^L(h^{t-1})$ is the s^L-component. We denote by $p_i(h^t)$ the probability of an observation sequence under opponent strategy i. We start by writing

$$\begin{aligned}
\mathbb{E}_i[N_i] - \mathbb{E}_j[N_i] &= \sum_{t=1}^{T} \mathbb{P}_i\left(S_t = e_i\right) - \mathbb{P}_j\left(S_t = e_i\right) \\
&= \sum_{h^{T-1}} \left(p_i(h^{T-1}) - p_j(h^{T-1})\right) \sum_{t=1}^{T} \mathbb{I}_{\{\mathcal{A}_t^s(h^{t-1})=e_i\}} \\
&\leq \sum_{h^{T-1}} \left(p_i(h^{T-1}) - p_j(h^{T-1})\right)^+ \sum_{t=1}^{T} \mathbb{I}_{\{\mathcal{A}_t^s(h^{t-1})=e_i\}} \\
&\leq T \sum_{h^{T-1}} \left(p_i(h^{T-1}) - p_j(h^{T-1})\right)^+ \\
&= \frac{T}{2} \|p_i(H^{T-1}) - p_j(H^{T-1})\|_1 \\
&\leq T \sqrt{\mathrm{KL}\left(p_j(H^{T-1}) \| p_i(H^{T-1})\right)/2}\,,
\end{aligned}$$

where $KL(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence, and in the last line we used Pinsker's inequality [Cover and Thomas, 2006, Lemma 17.3.2]. Now we upper bound $\mathrm{KL}\left(p_j(H^{t-1}) \| p_i(H^{t-1})\right)$ with the help of the chain-rule for KL-divergence [Cover and Thomas, 2006, Theorem 2.5.3].

$$\begin{aligned}
&\mathrm{KL}\left(p_j(H^{t-1}) \| p_i(H^{t-1})\right) \\
&= \sum_{t=1}^{T-1} \sum_{h^{t-1}} p_j(h^{t-1}) \sum_{h_t} p_j(h_t|h^{t-1}) \log \frac{p_j(h_t|h^{t-1})}{p_i(h_t|h^{t-1})} \\
&= \sum_{t=1}^{T-1} \sum_{h^{t-1}} p_j(h^{t-1}) \mathbb{I}_{\{\mathcal{A}_t^L(h^{t-1})=1\}} \sum_{h_t} p_j(h_t|h^{t-1}) \log \frac{p_j(h_t|h^{t-1})}{p_i(h_t|h^{t-1})}\,.
\end{aligned}$$

In the last line we used that if the algorithm does not request the label at time step t then $p_i(h_t|h^{t-1}) = p_j(h_t|h^{t-1})$. Observe that the last sum of the above expression

$$\sum_{h_t} p_j(h_t|h^{t-1}) \log \frac{p_j(h_t|h^{t-1})}{p_i(h_t|h^{t-1})} = \mathrm{KL}(p_j(H_t|h^{t-1}) \| p_i(H_t|h^{t-1}))\,,$$

and, by the data-processing inequality [Csiszár and Körner, 1981, Lemma 3.11],

$$\mathrm{KL}(p_j(H_t|h^{t-1}) \| p_i(H_t|h^{t-1})) \leq \mathrm{KL}(p_j(Z_t) \| p_i(Z_t))\,,$$

and thus

$$\begin{aligned}
\mathrm{KL}\left(p_j(H^{t-1}) \| p_i(H^{t-1})\right) &\leq \mathrm{KL}(p_j(Z_t) \| p_i(Z_t)) \sum_{t=1}^{T-1} \sum_{h^{t-1}} p_j(h^{t-1}) \mathbb{I}_{\{\mathcal{A}_t^L(h^{t-1})=1\}} \\
&= \mathrm{KL}(p_j(Z_t) \| p_i(Z_t)) \mathbb{E}_j[N_L]\,.
\end{aligned}$$

Finally, we upper bound $\mathrm{KL}(p_j(Z_t)\|p_i(Z_t))$ by $C_2 d\epsilon^2$ with the help of Lemma 7 from Chapter 6.[6] Putting everything together gives the statement of the lemma. □

[6]Note that by the statement of the lemma, the upper bound is looser by a factor of d. Nonetheless, in the proof the upper bound contains the minimal component of a d long probability vector, which can be upper bounded by $1/d$, gaining back the extra d factor.