Minimax Regret of Finite Partial-Monitoring Games in Stochastic Environments[*]

Gábor Bartók, Dávid Pál, Csaba Szepesvári
Department of Computing Science
University of Alberta
Edmonton, T6G 2E8, AB, Canada
{bartok,dpal,szepesva}@cs.ualberta.ca

Abstract

In a partial monitoring game, the learner repeatedly chooses an action, the environment responds with an outcome, and then the learner suffers a loss and receives a feedback signal, both of which are fixed functions of the action and the outcome. The goal of the learner is to minimize his regret, which is the difference between his total cumulative loss and the total loss of the best fixed action in hindsight. Assuming that the outcomes are generated in an i.i.d. fashion from an arbitrary and unknown probability distribution, we characterize the minimax regret of any partial monitoring game with finitely many actions and outcomes. It turns out that the minimax regret of any such game is either zero, $\widetilde{\Theta}(\sqrt{T})$, $\Theta(T^{2/3})$, or $\Theta(T)$. We provide a computationally efficient learning algorithm that achieves the minimax regret within logarithmic factor for any game.

1 Introduction

Partial monitoring provides a mathematical framework for sequential decision making problems with imperfect feedback. Various problems of interest can be modeled as partial monitoring instances, such as learning with expert advice (Littlestone and Warmuth, 1994), the multi-armed bandit problem (Auer et al., 2002), dynamic pricing (Kleinberg and Leighton, 2003), the dark pool problem (Agarwal et al., 2010), label efficient prediction (Cesa-Bianchi et al., 2005), and linear and convex optimization with full or bandit feedback (Zinkevich, 2003, Abernethy et al., 2008, Flaxman et al., 2005).

In this paper we restrict ourselves to finite games, *i.e.,* games where both the set of actions available to the learner and the set of possible outcomes generated by the environment are finite. A finite partial monitoring game $\mathbf{G}$ is described by a pair of $N \times M$ matrices: the *loss matrix* $\mathbf{L}$ and the *feedback matrix* $\mathbf{H}$. The entries $\ell_{i,j}$ of $\mathbf{L}$ are real numbers lying in, say, the interval $[0,1]$. The entries $h_{i,j}$ of $\mathbf{H}$ belong to an alphabet Σ on which we do not impose any structure and we only assume that learner is able to distinguish distinct elements of the alphabet.

The game proceeds in T rounds according to the following protocol. First, $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ is announced for both players. In each round $t = 1, 2, \ldots, T$, the learner chooses an action $I_t \in \{1, 2, \ldots, N\}$ and simultaneously, the environment chooses an outcome $J_t \in \{1, 2, \ldots, M\}$. Then, the learner receives as a feedback the entry h_{I_t,J_t}. The learner incurs *instantaneous loss* ℓ_{I_t,J_t}, which is *not revealed* to him. The feedback can be thought of as a masked information about the outcome J_t. In some cases h_{I_t,J_t} might uniquely determine the outcome, in other cases the feedback might give only partial or no information about the outcome. In this paper, we shall assume that J_t is chosen randomly from a fixed multinomial distribution.

The learner is scored according to the loss matrix $\mathbf{L}$. In round t the learner incurs an *instantaneous loss* of ℓ_{I_t,J_t}. The goal of the learner is to keep low his *total loss* $\sum_{t=1}^{T} \ell_{I_t,J_t}$. Equivalently, the learner's performance can also be measured in terms of his regret, *i.e.,* the total loss of the learner is compared with the loss of best fixed action in hindsight. The regret is defined as the difference of these two losses.

In general, the regret grows with the number of rounds T. If the regret is sublinear in T, the learner is said to be Hannan consistent, and this means that the learner's average per-round loss approaches the average per-round loss of the best action in hindsight.

Piccolboni and Schindelhauer (2001) were one of the first to study the regret of these games. In fact, they have studied the problem without making any probabilistic assumptions about the outcome sequence J_t.

[*] This work was supported in part by AICML, AITF (formerly iCore and AIF), NSERC and the PASCAL2 Network of Excellence under EC grant no. 216886.

They proved that for any finite game $(\mathbf{L}, \mathbf{H})$, either for any algorithm the regret can be $\Omega(T)$ in the worst case, or there exists an algorithm which has regret $\widetilde{O}(T^{3/4})$ on any outcome sequence[1]. This result was later improved by Cesa-Bianchi et al. (2006) who showed that the algorithm of Piccolboni and Schindelhauer has regret $O(T^{2/3})$. Furthermore, they provided an example of a finite game, a variant of label-efficient prediction, for which any algorithm has regret $\Theta(T^{2/3})$ in the worst case.

However, for many games $O(T^{2/3})$ is not optimal. For example, games with full feedback (*i.e.,* when the feedback uniquely determines the outcome) can be viewed as a special instance of the problem of learning with expert advice and in this case it is known that the "EWA forecaster" has regret $O(\sqrt{T})$; see *e.g.,* Lugosi and Cesa-Bianchi (2006, Chapter 3). Similarly, for games with "bandit feedback" (*i.e.,* when the feedback determines the instantaneous loss) the INF algorithm (Audibert and Bubeck, 2009) and the Exp3 algorithm (Auer et al., 2002) achieve $O(\sqrt{T})$ regret as well.[2]

This leaves open the problem of determining the minimax regret (*i.e.,* optimal worst-case regret) of any given game $(\mathbf{L}, \mathbf{H})$. A partial progress was made in this direction by Bartók et al. (2010) who characterized (almost) all finite games with $M = 2$ outcomes. They showed that the minimax regret of any "non-degenerate" finite game with two outcomes falls into one of four categories: zero, $\widetilde{\Theta}(\sqrt{T})$, $\Theta(T^{2/3})$ or $\Theta(T)$. They gave a combinatoric-geometric condition on the matrices $\mathbf{L}, \mathbf{H}$ which determines the category a game belongs to. Additionally, they constructed an efficient algorithm which, for any game, achieves the minimax regret rate associated to the game within poly-logarithmic factor.

In this paper, we consider the same problem, with two exceptions. In pursuing a general result, we will consider *all* finite games. However, at the same time, we will only deal with *stochastic* environments, *i.e.,* when the outcome sequences are generated from a fixed probability distribution in an i.i.d. manner.

The regret against stochastic environments is defined as the difference between the cumulative loss suffered by the algorithm and that of the action with the lowest expected loss. That is, given an algorithm $\mathcal{A}$ and a time horizon T, if the outcomes are generated from a probability distribution p, the regret is

$$R_T(\mathcal{A}, p) = \sum_{t=1}^{T} \ell_{I_t, J_t} - \min_{1 \le i \le N} \mathbb{E}_p \left[\sum_{t=1}^{T} \ell_{i, J_t} \right] .$$

In this paper we analyze the *minimax* expected regret (in what follows, minimax regret) of games, defined as

$$R_T(\mathbf{G}) = \inf_{\mathcal{A}} \sup_{p \in \Delta_M} \mathbb{E}_p \left[R_T(\mathcal{A}, p) \right] .$$

We show that the minimax regret of any finite game falls into four categories: zero, $\widetilde{\Theta}(\sqrt{T})$, $\Theta(T^{2/3})$, or $\Theta(T)$. Accordingly, we call the games *trivial*, *easy*, *hard*, and *hopeless*. We give a simple and efficiently computable characterization of these classes using a geometric condition on $(\mathbf{L}, \mathbf{H})$. We provide lower-bounds and algorithms that achieve them within poly-logarithmic factor. Our result is an extension of the result of Bartók et al. (2010) for stochastic environments.

It is clear that any lower bound which holds for stochastic environments must hold for adversarial environments too. On the other hand, algorithms and regret upper bounds for stochastic environments, of course, do not transfer to algorithms and regret upper bounds for the adversarial case. Our characterization is a stepping stone towards understanding the minimax regret of partial monitoring games. In particular, we conjecture that our characterization holds without any change for unrestricted environments.

2 Preliminaries

In this section, we introduce our conventions, along with some definitions. By default, all vectors are column vectors. We denote by $\|v\| = \sqrt{v^\top v}$ the Euclidean norm of a vector v. For a vector v, the notation $v \ge 0$ means that all entries of v are non-negative, and the notation $v > 0$ means that all entries are positive. For a matrix A, $\operatorname{Im} A$ denotes its *image space*, *i.e.,* the vector space generated by its columns, and the notation $\operatorname{Ker} A$ denotes its *kernel*, *i.e.,* the set $\{x : Ax = 0\}$.

Consider a game $\mathbf{G} = (\mathbf{L}, \mathbf{H})$ with N actions and M outcomes. That is, $\mathbf{L} \in \mathbb{R}^{N \times M}$ and $\mathbf{H} \in \Sigma^{N \times M}$. For the sake of simplicity and, without loss of generality, we assume that no symbol $\sigma \in \Sigma$ can be present in two different rows of $\mathbf{H}$. The *signal matrix* of an action is defined as follows:

Definition 1 (Signal matrix) *Let $\{\sigma_1, \dots, \sigma_{s_i}\}$ be the set of symbols listed in the i^{th} row of $\mathbf{H}$. (Thus, s_i denotes the number of different symbols in row i of $\mathbf{H}$). The* signal matrix S_i *of action i is defined as an*

[1] The notations $\widetilde{O}(\cdot)$ and $\widetilde{\Theta}(\cdot)$ hide polylogarithmic factors.

[2] We ignore the dependence of regret on the number of actions or any other parameters.

$s_i \times M$ matrix with entries $a_{k,j} = \mathbb{I}(h_{i,j} = \sigma_k)$ for $1 \le k \le s_i$ and $1 \le j \le M$. The signal matrix for a set of actions is defined as the signal matrices of the actions in the set, stacked on top of one another, in the ordering of the actions.

For an example of a signal matrix, see Section 3.1. We identify the strategy of a stochastic opponent with an element of the probability simplex $\Delta_M = \{p \in \mathbb{R}^M \ : \ p \ge 0, \ \sum_{j=1}^M p_j = 1\}$. Note that for any opponent strategy p, if the learner chooses action i then the vector $S_i p \in \mathbb{R}^{s_i}$ is the probability distribution of the observed feedback: $(S_i p)_k$ is the probability of observing the k^{th} symbol.

We denote by $\ell_i^\top$ the i^{th} row of the loss matrix $\mathbf{L}$ and we call ℓ_i the *loss vector of action i*. We say that action i is *optimal* under opponent strategy $p \in \Delta_M$ if for any $1 \le j \le N$, $\ell_i^\top p \le \ell_j^\top p$. Action i is said to be *Pareto-optimal* if there exists an opponent strategy p such that action i is optimal under p. We now define the *cell decomposition* of Δ_M induced by $\mathbf{L}$ (for an example, see Figure 2):

Definition 2 (Cell decomposition) *For an action i, the* cell *C_i associated with i is defined as $C_i = \{p \in \Delta_M \ : \ \text{action } i \text{ is optimal under } p\}$. The* cell decomposition *of Δ_M is defined as the multiset $\mathcal{C} = \{C_i \ : \ 1 \le i \le N, \ C_i \text{ has positive } (M-1)\text{-dimensional volume}\}$.*

Actions whose cell is of positive $(M-1)$-dimensional volume are called *strongly Pareto-optimal*. Actions that are Pareto-optimal but not strongly Pareto-optimal are called *degenerate*. Note that the cells of the actions are defined with linear inequalities and thus they are convex polytopes. It follows that strongly Pareto-optimal actions are the actions whose cells are $(M-1)$-dimensional polytopes. It is also important to note that the cell decomposition is a multiset, since some actions can share the same cell. Nevertheless, if two actions have the same cell of dimension $(M-1)$, their loss vectors will necessarily be identical.[3]

We call two cells of $\mathcal{C}$ *neighbors* if their intersection is an $(M-2)$-dimensional polytope. The actions corresponding to these cells will also be called neighbors. Neighborship is not defined for cells outside of $\mathcal{C}$. For two neighboring cells $C_i, C_j \in \mathcal{C}$, we define the *neighborhood action set* $A_{i,j} = \{1 \le k \le N \ : \ C_i \cap C_j \subseteq C_k\}$. It follows from the definition that actions i and j are in $A_{i,j}$ and thus $A_{i,j}$ is nonempty. However, one can have more than two actions in the neighborhood action set.

When discussing lower bounds we will need the definition of algorithms. For us, an algorithm $\mathcal{A}$ is a mapping $\mathcal{A} : \Sigma^* \to \{1, 2, \ldots, N\}$ which 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.

3 Classification of finite partial-monitoring games

In this section we present our main result: we state the theorem that classifies all finite stochastic partial-monitoring games based on how their minimax regret scales with the time horizon. Thanks to the previous section, we are now equipped to define a notion which will play a key role in the classification theorem:

Definition 3 (Observability) *Let S be the signal matrix for the set of all actions in the game. For actions i and j, we say that $\ell_i - \ell_j$ is* globally observable *if $\ell_i - \ell_j \in \operatorname{Im} S^\top$. Furthermore, if i and j are two neighboring actions, then $\ell_i - \ell_j$ is called* locally observable *if $\ell_i - \ell_j \in \operatorname{Im} S_{(i,j)}^\top$, where $S_{(i,j)}$ is the signal matrix for the neighborhood action set $A_{i,j}$.*

As we will see, global observability implies that we can estimate the difference of the expected losses after choosing each action once. Local observability means we only need actions from the neighborhood action set to estimate the difference.

The classification theorem, which is our main result, is the following:

Theorem 4 (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 strongly 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 for all pairs of (strongly Pareto-optimal) neighboring actions i and j, $\ell_i - \ell_j$ is locally observable. These games are called* easy.

[3] One could think that actions with identical loss vectors are redundant and that all but one of such actions could be removed without loss of generality. However, since different actions can lead to different observations and thus yield different information, removing the duplicates can be harmful.

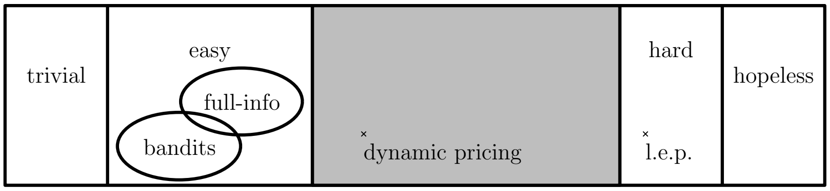

Figure 1: Partial monitoring games and their minimax regret as it was known previously. The big rectangle denotes the set of all games. Inside the big rectangle, the games are ordered from left to right based on their minimax regret. In the "hard" area, l.e.p. denotes label-efficient prediction. The 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.

(d) $R_T(\mathbf{G}) = \Theta(T^{2/3})$ *if* $\mathbf{G}$ *is not hopeless and there exists a pair of neighboring actions* i *and* j *such that* $\ell_i - \ell_j$ *is* not *locally observable. These games are called* 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 remainder of the paper is dedicated to proving Theorem 4. 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[4]. 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) was proved by Antos et al. (2011). For a visualization of previous results, see Figure 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 paper 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"[5]. This algorithm is presented in Section 4, while its analysis is given in Section 5. The lower bound for case (d) is presented in Section 6.

3.1 Example

In this section, as a corollary of Theorem 4 we show that the discretized dynamic pricing game (see, *e.g.,* Cesa-Bianchi et al. (2006)) is *hard.* Dynamic pricing is a game between a vendor (learner) and a customer (environment). 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 (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 2 for the cell-decomposition for $N = 3$). It is easy to see that all the actions are strongly Pareto-optimal. Also, after some linear algebra it turns out that the cells underlying the actions have a single common vertex in the interior of the probability simplex. It follows that any two actions are neighbors. On the other hand, if we take two non-consecutive actions i and i', $\ell_i - \ell_{i'}$ is not

[4] 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).

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

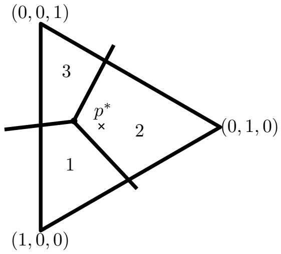

Figure 2: The cell decomposition of the discretized dynamic pricing game with 3 actions. If the opponent strategy is p^*, then action 2 is the optimal action.

locally observable. For example, the signal matrix for action 1 and action N is

$$S_{(1,N)} = \begin{pmatrix} 1 & \dots & 1 & 1 \\ 1 & \dots & 1 & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix} ,$$

whereas $\ell_N - \ell_1 = (c, c-1, \dots, c-N+2, -N+1)^\top$. It is obvious that $\ell_N - \ell_1$ is not in the row space of $S_{(1,N)}$.

4 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 4). 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 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 cells C_i, C_j, the signal matrix $S_{(i,j)}$ is defined as the signal matrix for the neighborhood action set $A_{i,j}$ of cells i, j. Assuming that the game satisfies the condition of case (c) of Theorem 4, we have that for all neighboring cells C_i and C_j, $\ell_i - \ell_j \in \operatorname{Im} S_{(i,j)}^\top$. This means that there exists a *coefficient vector* $v_{(i,j)}$ such that $\ell_i - \ell_j = S_{(i,j)}^\top v_{(i,j)}$. We define the k^{th} *segment* of $v_{(i,j)}$, denoted by $v_{(i,j),k}$, as the vector of components of $v_{(i,j)}$ that correspond to the k^{th} action in the neighborhood action set. That is, if $S_{(i,j)}^\top = \begin{pmatrix} S_1^\top & \cdots & S_r^\top \end{pmatrix}$, then $\ell_i - \ell_j = S_{(i,j)}^\top v_{(i,j)} = \sum_{s=1}^r S_s^\top v_{(i,j),s}$, where $S_1, \dots, S_r$ are the signal matrices of the individual actions in $A_{i,j}$.

Let $J_t \in \{1, \dots, M\}$ denote the outcome at time step t. For $1 \le k \le M$, let $e_k \in \mathbb{R}^M$ be the k^{th} unit vector. For an action i, let $O_i(t) = S_i e_{J_t}$ be the *observation vector* of action i at time step t. If the rows of the signal matrix S_i correspond to symbols $\sigma_1, \dots, \sigma_{s_i}$ and action i is chosen at time step t then the unit vector $O_i(t)$ indicates which symbol was observed in that time step. Thus, $O_{I_t}(t)$ holds the same information as the feedback at time t (recall that I_t is the action chosen by the learner at time step t). From now on, for simplicity, we will assume that the feedback at time step t is the observation vector $O_{I_t}(t)$ itself.

The main idea of the algorithm is to successively eliminate actions in an efficient, yet safe manner. When all remaining strongly 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 each 'alive' Pareto optimal 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 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 signal matrices $S_{(i,j)}$ assigned to the edges of the graph, the coefficient vectors $v_{(i,j)}$ and their segment 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 $\hat{\mu}_{(i,j)}$ of the loss-differences $(\ell_i - \ell_j)^\top p^*$ are updated, where p^* is the actual opponent strategy. The algorithm maintains the set $\mathcal{C}$ of cells of alive actions and their neighborship graph $\mathcal{G}$.

Algorithm 1 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 $\mathcal{G}$: $\mu_{(i,j)} \leftarrow \sum_{k \in A_{i,j}} O_k^\top v_{(i,j),k}$ end for
    for each non-adjacent vertex pair $(i,j)$ in $\mathcal{G}$: $\mu_{(i,j)} \leftarrow \sum_{(k,l) \in path_{(i,j)}} \mu_{(k,l)}$ end for
    $haveEliminated \leftarrow$ false
    for each vertex pair $(i,j)$ in $\mathcal{G}$ do
        $\hat{\mu}_{(i,j)} \leftarrow \left(1 - \frac{1}{n}\right) \hat{\mu}_{(i,j)} + \frac{1}{n} \mu_{(i,j)}$
        if BSTOPSTEP$(\hat{\mu}_{(i,j)}, LB_{(i,j)}, UB_{(i,j)}, \sigma_{(i,j)}, R_{(i,j)}, n, 1/2, \delta)$ then
            $[aliveActions, \mathcal{C}, \mathcal{G}] \leftarrow$ ELIMINATE$(i, j, \mathrm{sgn}(\hat{\mu}_{(i,j)}))$
            $haveEliminated \leftarrow$ true
        end if
    end for
    if $haveEliminated$ then
        $\{path_{(i,j)}\} \leftarrow$ REGENERATEPATHS$(\mathcal{G})$
    end if
end while
Let $i$ be a strongly Pareto-optimal action in $aliveActions$
while $t < T$ do
    EXECUTEACTION(i)
    $t \leftarrow t + 1$
end while
```

The estimates are calculated as follows. First we calculate estimates for neighboring actions (i,j). In round[6] n, for every action k in $A_{i,j}$ let O_k be the observation vector for action k. Let $\mu_{(i,j)} = \sum_{k \in A_{i,j}} O_k^\top v_{(i,j),k}$. From the local observability condition and the construction of $v_{(i,j),k}$, with simple algebram it follows that $\mu_{(i,j)}$ are unbiased estimates of $(\ell_i - \ell_j)^\top p^*$ (see Lemma 5). 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 $\mu_{(i,j)}(n)$ will be the sum of the estimates along the path: $\sum_{l=1}^{r} \mu_{(i_{l-1}, i_l)}$. The estimate of the difference of the expected losses after round n will be the average $\hat{\mu}_{(i,j)} = (1/n) \sum_{l=1}^{n} \mu_{(i,j)}(s)$, where $\mu_{(i,j)}(s)$ 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 $\hat{\mu}_{(i,j)}$ and its relative error. This subroutine uses a stopping rule based on Bernstein's inequality.

The subroutine's pseudocode is shown as Algorithm 2 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

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

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

[6] 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.

Algorithm 2 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** $(1+\epsilon)\mathrm{LB} < (1-\epsilon)\mathrm{UB}$
```

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 $\hat{\mu}$ 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 11 (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 (1) influences only C.

If BSTOPSTEP returns true for an estimate $\mu_{(i,j)}$, function ELIMINATE is called. If, say, $\mu_{(i,j)} > 0$, 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 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.

5 Analysis of the algorithm

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

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 $\alpha_{i,j} = (\ell_i - \ell_j)^\top p^*$ be the expected difference of their instantaneous loss. The expected regret of the algorithm can be rewritten as

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_{I_t,J_t} - \min_{1\leq i\leq N} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{i,J_t}\right]\right] = \sum_{i=1}^{N} \mathbb{E}\left[\tau_i\right] \alpha_{i,j^*}, \tag{2}$$

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 $\{1, \ldots, M\}$, 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

$$J_t = J_{I_t}\left(\textstyle\sum_{s=1}^{t} \mathbb{I}(I_s = I_t)\right). \tag{3}$$

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 (3). As we will see, this construction significantly simplifies subsequent steps of the proof. In particular, the construction will be very convenient since if action k is selected by our algorithm in the n^{th} elimination round then the outcome obtained in response is going to be $O_k(n) = S_k u_k(n)$, 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 $\mu_{(i,j)}(n)$, the estimate of $\alpha_{i,j}$ obtained in round n is $\mathcal{F}_n$-measurable. The next lemma establishes that, furthermore, $\mu_{(i,j)}(n)$ is an unbiased estimate of $\alpha_{i,j}$:

Lemma 5 *For any $n \geq 1$ and i, j such that $C_i, C_j \in \mathcal{C}$, $\mathbb{E}_{n-1}[\mu_{(i,j)}(n)] = \alpha_{i,j}$.*

Proof: Consider first the case when actions i and j are neighbors. In this case,

$$\mu_{(i,j)}(n) = \sum_{k \in A_{i,j}} O_k(n)^\top v_{(i,j),k} = \sum_{k \in A_{i,j}} (S_k u_k(n))^\top v_{(i,j),k} = \sum_{k \in A_{i,j}} u_k(n)^\top S_k^\top v_{(i,j),k}\,,$$

and thus

$$\begin{aligned}\mathbb{E}_{n-1}\left[\mu_{(i,j)}(n)\right] &= \sum_{k \in A_{i,j}} \mathbb{E}_{n-1}\left[u_k(n)^\top\right] S_k^\top v_{(i,j),k} = {p^*}^\top \sum_{k \in A_{i,j}} S_k^\top v_{(i,j),k} = {p^*}^\top S_{(i,j)}^\top v_{(i,j)} \\ &= {p^*}^\top (\ell_i - \ell_j) = \alpha_{i,j}\,.\end{aligned}$$

For non-adjacent i and j, we have a telescoping sum:

$$\mathbb{E}_{n-1}\left[\mu_{(i,j)}(n)\right] = \sum_{k=1}^{r} \mathbb{E}_{n-1}[\mu_{(i_{k-1},i_k)}(n)] = {p^*}^\top \left(\ell_{i_0} - \ell_{i_1} + \ell_{i_1} - \ell_{i_2} + \cdots + \ell_{i_{r-1}} - \ell_{i_r}\right) = \alpha_{i,j}\,,$$

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 6 *The conditional variance of $\mu_{(i,j)}(n)$, $\mathbb{V}\mathrm{ar}_{n-1}(\mu_{(i,j)}(n))$, is upper bounded by $V = 2\sum_{\{i,j \text{ neighbors}\}} \|v_{(i,j)}\|_2^2$.*

Proof: For neighboring cells i, j, we write

$$\mu_{(i,j)}(n) = \sum_{k \in A_{i,j}} O_k(n)^\top v_{(i,j),k} \qquad \text{and thus}$$

$$\begin{aligned}\mathbb{V}\mathrm{ar}_{n-1}(\mu_{(i,j)}(n)) &= \mathbb{V}\mathrm{ar}_{n-1}\left(\sum_{k \in A_{i,j}} O_k(n)^\top v_{(i,j),k}\right) \\ &= \sum_{k \in A_{i,j}} \mathbb{E}_{n-1}\left[v_{(i,j),k}^\top (O_k(n) - \mathbb{E}_{n-1}[O_k(n)])(O_k(n) - \mathbb{E}_{n-1}[O_k(n)])^\top v_{(i,j),k}\right] \\ &\leq \sum_{k \in A_{i,j}} \|v_{(i,j),k}\|_2^2\, \mathbb{E}_{n-1}\left[\|O_k(n) - \mathbb{E}_{n-1}[O_k(n)]\|_2^2\right] \\ &\leq \sum_{k \in A_{i,j}} \|v_{(i,j),k}\|_2^2 = \|v_{(i,j)}\|_2^2\,, \end{aligned} \tag{4}$$

where in (4) we used that $O_k(n)$ is a unit vector and $\mathbb{E}_{n-1}[O_k(n)]$ 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 $\mu_{(i,j)}(n)$ can be written as

$$\mu_{(i,j)}(n) = \sum_{s=1}^{r} \mu_{(i_{s-1},i_s)}(n) = \sum_{s=1}^{r} \sum_{k \in A_{i_{s-1},i_s}} O_k(n)^\top v_{(i_{s-1},i_s),k}\,.$$

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

$$\sum_{k \in \bigcup A_{i_{s-1},i_s}} O_k(n)^\top (v_{(i_{s_k-1},i_{s_k}),k} + v_{(i_{s_k} i_{s_k+1}),k})\,,$$

and thus $\mathbb{V}\mathrm{ar}_{n-1}\left(\mu_{(i,j)}(n)\right) \leq 2\sum_{s=1}^{r} \|v_{(i_{s-1},i_s)}\|_2^2 \leq 2\sum_{\{i,j \text{ neighbors}\}} \|v_{(i,j)}\|_2^2$. ■

Lemma 7 *The range of the estimates $\mu_{(i,j)}(n)$ is upper bounded by $R = \sum_{\{i,j \text{ neighbors}\}} \|v_{(i,j)}\|_1$.*

Proof: The bound trivially follows from the definition of the estimates. ■

Let δ be the confidence parameter used in BStopStep. Since, according to Lemmas 5, 6 and 7, $(\mu_{(i,j)})$ is a "shifted" martingale difference sequence with conditional mean $\alpha_{i,j}$, bounded conditional variance and range, we can apply Lemma 11 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 $\operatorname{sgn}(\alpha_{i,j}) = \operatorname{sgn}(\mu_{(i,j)})$ and the algorithm eliminates all the actions whose cell is contained in the closed half space defined by $\mathcal{H} = \{p \ : \ \operatorname{sgn}(\alpha_{i,j})p^\top(\ell_i - \ell_j) \le 0\}$. By definition $\alpha_{i,j} = (\ell_i - \ell_j)^\top p^*$. Thus $p^* \notin \mathcal{H}$ and none of the eliminated actions can be optimal under p^*.

From Lemma 11 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

$$\tau_i^* \le \frac{c(\mathbf{G})}{\alpha_{i,j^*}^2} \log \frac{R}{\delta \alpha_{i,j^*}} = T_i \,, \tag{5}$$

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

The following lemma, the proof of which can be found in the Appendix, shows that degenerate actions will be eliminated in time.

Lemma 8 *Let action i be a degenerate action. Let $A_i = \{j \ : \ C_j \in \mathcal{C}, C_i \subset C_j\}$. The following two statements hold:*

1. *If any of the actions in A_i is eliminated, then action i is eliminated as well.*
2. *There exists an action $k_i \in A_i$ such that $\alpha_{k_i,j^*} \ge \alpha_{i,j^*}$.*

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 then that of action k_i. Hence, $\tau_i^* \le \tau_{k_i}^*$.

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

$$\begin{aligned}
\sum_{i=1}^N \mathbb{E}\left[\tau_i\right] \alpha_{i,j^*} &= \sum_{i=1}^N \mathbb{E}\left[\mathbb{I}(\mathcal{E})\tau_i\right] \alpha_{i,j^*} + \mathbb{P}(\mathcal{E}^c) T \qquad \left(\text{because} \textstyle\sum_{i=1}^N \tau_i = T \text{ and } 0 \le \alpha_{i,j^*} \le 1\right) \\
&\le \sum_{i=1}^N \mathbb{E}\left[\mathbb{I}(\mathcal{E})\tau_i^*\right] \alpha_{i,j^*} + N^2\delta T \\
&\le \sum_{i:\ C_i \in \mathcal{C}} \mathbb{E}\left[\mathbb{I}(\mathcal{E})\tau_i^*\right] \alpha_{i,j^*} + \sum_{i:\ C_i \notin \mathcal{C}} \mathbb{E}\left[\mathbb{I}(\mathcal{E})\tau_i^*\right] \alpha_{i,j^*} + N^2\delta T \\
&\le \sum_{i:\ C_i \in \mathcal{C}} \mathbb{E}\left[\mathbb{I}(\mathcal{E})\tau_i^*\right] \alpha_{i,j^*} + \sum_{i:\ C_i \notin \mathcal{C}} \mathbb{E}\left[\mathbb{I}(\mathcal{E})\tau_{k_i}^*\right] \alpha_{k_i,j^*} + N^2\delta T \qquad \text{(by Lemma 8)} \\
&\le \sum_{i:\ C_i \in \mathcal{C}} T_i \alpha_{i,j^*} + \sum_{i:\ C_i \notin \mathcal{C}} T_{k_i} \alpha_{k_i,j^*} + N^2\delta T \\
&\le \sum_{\substack{i:\ C_i \in \mathcal{C} \\ \alpha_{i,j^*} \ge \alpha_0}} T_i \alpha_{i,j^*} + \sum_{\substack{i:\ C_i \notin \mathcal{C} \\ \alpha_{k_i,j^*} \ge \alpha_0}} T_{k_i} \alpha_{k_i,j^*} + \left(\alpha_0 + N^2\delta\right) T \\
&\le c(\mathbf{G}) \left(\sum_{\substack{i:\ C_i \in \mathcal{C} \\ \alpha_{i,j^*} \ge \alpha_0}} \frac{\log \frac{R}{\delta \alpha_{i,j^*}}}{\alpha_{i,j^*}} + \sum_{\substack{i:\ C_i \notin \mathcal{C} \\ \alpha_{k_i,j^*} \ge \alpha_0}} \frac{\log \frac{R}{\delta \alpha_{k_i,j^*}}}{\alpha_{k_i,j^*}} \right) + \left(\alpha_0 + N^2\delta\right) T \\
&\le c(\mathbf{G}) N \frac{\log \frac{R}{\delta \alpha_0}}{\alpha_0} + \left(\alpha_0 + N^2\delta\right) T \,,
\end{aligned}$$

The above calculation holds for any value of $\alpha_0 > 0$. Setting

$$\alpha_0 = \sqrt{\frac{c(\mathbf{G})N}{T}} \qquad \text{and} \qquad \delta = \sqrt{\frac{c(\mathbf{G})}{TN^3}}\,, \qquad \text{we get}$$

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

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 A lower bound for hard games

In this section we prove that for any game that satisfies the condition of Case (d) of Theorem 4, the minimax regret is of $\Omega(T^{2/3})$.

Theorem 9 *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 such that $\ell_i - \ell_j \notin \operatorname{Im} S_{(i,j)}^\top$. 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*

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

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 $\mathcal{L}_3$ collect the loss vectors of actions which lie on the open segment connecting ℓ_1 and ℓ_2. It is easy to see that $\mathcal{L}_3$ is the set of loss vectors that correspond to the cell C_3. We define $\mathcal{L}_4$ as the set of all the other loss vectors. For $i = 1, 2, 3, 4$, let $k_i = |A_i|$.

Let $S = S_{i,j}$ the signal matrix of the neighborhood action set of C_1 and C_2. It follows from the assumption of the theorem that $\ell_2 - \ell_1 \notin \operatorname{Im}(S^\top)$. Thus, $\{\rho(\ell_2 - \ell_1) \ : \ \rho \in \mathbb{R}\} \not\subset \operatorname{Im}(S^\top)$, or equivalently, $(\ell_2 - \ell_1)^\perp \not\supset \operatorname{Ker} S$, where we used that $(\operatorname{Im} M)^\perp = \operatorname{Ker}(M^\top)$. Thus, there exists a vector v such that $v \in \operatorname{Ker} S$ and $(\ell_2 - \ell_1)^\top v \neq 0$. By scaling we can assume that $(\ell_2 - \ell_1)^\top v = 1$. Note that since $v \in \operatorname{Ker} S$ and the rowspace of S 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 $\mathcal{L}_3$ we know that for any $\ell \in \mathcal{L}_3$, $\ell - \ell_1 = \eta_\ell(\ell_2 - \ell_1)$ and $\ell - \ell_2 = (1 - \eta_\ell)(\ell_1 - \ell_2)$ for some $0 < \eta_\ell < 1$. Let $\lambda_1 = \min_{\ell \in \mathcal{L}_3} \eta_\ell$ and $\lambda_2 = \min_{\ell \in \mathcal{L}_3}(1 - \eta_\ell)$ and $\lambda = \min(\lambda_1, \lambda_2)$ if $\mathcal{L}_3 \neq \emptyset$ and let $\lambda = 1/2$, otherwise. Finally, let $\beta_i = \min_{\ell \in \mathcal{L}_4}(\ell - \ell_i)^\top p_i$ 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:

$$\begin{aligned}
R_T^{(1)} &\geq N_2^1 \overbrace{(\ell_2 - \ell_1)^\top p_1}^{\varepsilon} + N_3^1 \lambda (\ell_2 - \ell_1)^\top p_1 + N_4^1 \beta \geq \lambda(N_2^1 + N_3^1)\varepsilon + N_4^1 \beta\,, \\
R_T^{(2)} &\geq N_1^2 \underbrace{(\ell_1 - \ell_2)^\top p_2}_{\varepsilon} + N_3^2 \lambda (\ell_1 - \ell_2)^\top p_2 + N_4^2 \beta \geq \lambda(N_1^2 + N_3^2)\varepsilon + N_4^2 \beta\,.
\end{aligned} \tag{6}$$

For the next step, we need the following lemma.

Lemma 10 *There exists a (problem dependent) constant c such that the following inequalities hold:*

$$N_1^2 \geq N_1^1 - cT\varepsilon\sqrt{N_4^1}\,, \qquad N_3^2 \geq N_3^1 - cT\varepsilon\sqrt{N_4^1}\,,$$

$$N_2^1 \geq N_2^2 - cT\varepsilon\sqrt{N_4^2}\,, \qquad N_3^1 \geq N_3^2 - cT\varepsilon\sqrt{N_4^2}\,.$$

Proof: (Lemma 10) 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 $k \in \{1,2,3\}$,

$$\begin{aligned}
N_k^i - N_k^j &= \sum_{f^{T-1}} \left(p_i^*(f^{T-1}) - p_j^*(f^{T-1})\right) \sum_{t=0}^{T-1} \mathbb{I}(\mathcal{A}(f^t) \in A_k) \\
&\leq \sum_{\substack{f^{T-1}: \\ p_i^*(f^{T-1}) - p_j^*(f^{T-1}) \geq 0}} \left(p_i^*(f^{T-1}) - p_j^*(f^{T-1})\right) \sum_{t=0}^{T-1} \mathbb{I}(\mathcal{A}(f^t) \in A_k) \\
&\leq T \sum_{\substack{f^{T-1}: \\ p_i^*(f^{T-1}) - p_j^*(f^{T-1}) \geq 0}} p_i^*(f^{T-1}) - p_j^*(f^{T-1}) = \frac{T}{2} \|p_1^* - p_2^*\|_1 \\
&\leq T \sqrt{\mathrm{KL}(p_1^* || p_2^*)/2}\,, \qquad (7)
\end{aligned}$$

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

$$\begin{aligned}
\mathrm{KL}(p_1^*||p_2^*) &= \sum_{t=1}^{T-1} \sum_{f^{t-1}} p_1^*(f^{t-1}) \sum_{f_t} p_1^*(f_t|f^{t-1}) \log \frac{p_1^*(f_t|f^{t-1})}{p_2^*(f_t|f^{t-1})} \\
&= \sum_{t=1}^{T-1} \sum_{f^{t-1}} p_1^*(f^{t-1}) \sum_{i=1}^{4} \mathbb{I}(\mathcal{A}(f^{t-1}) \in A_i) \sum_{f_t} p_1^*(f_t|f^{t-1}) \log \frac{p_1^*(f_t|f^{t-1})}{p_2^*(f_t|f^{t-1})} \qquad (8)
\end{aligned}$$

Let $a_{f_t}^\top$ be the row of S that corresponds to the feedback symbol f_t.[7] 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 (8) by writing

$$\begin{aligned}
\mathrm{KL}(p_1^*||p_2^*) &\leq \sum_{t=1}^{T-1} \sum_{f^{t-1}} p_1^*(f^{t-1}) \mathbb{I}(\mathcal{A}(f^{t-1}) \in A_4) \sum_{f_t} p_1^*(f_t|f^{t-1}) \log \frac{p_1^*(f_t|f^{t-1})}{p_2^*(f_t|f^{t-1})} \\
&\leq c_1 \varepsilon^2 \sum_{t=1}^{T-1} \sum_{f^{t-1}} p_1^*(f^{t-1}) \mathbb{I}(\mathcal{A}(f^{t-1}) \in A_4) \qquad (9) \\
&\leq c_1 \varepsilon^2 N_4^1\,.
\end{aligned}$$

In (9) we used Lemma 12 (see Appendix) to upper bound the KL-divergence of p_1 and p_2. Flipping p_1^* and p_2^* in (7) we get the same result with N_4^2. Reading together with the bound in (7) we get all the desired inequalities. ■

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 10. If $i = r$ then it is a trivial lower bounding by subtracting a positive value. From (6) we have

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

[7] Recall that we assumed that different actions have difference feedback symbols, and thus a row of S corresponding to a symbol is unique.

Now assume that, at the beginning of the game, the opponent randomly chooses between strategies p_1 and p_2 with equal probability. The 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 12 is satisfied, completing the proof of Theorem 9. ■

7 Discussion

In this we paper we classified all finite partial-monitoring games under stochastic environments, based on their minimax regret. We conjecture that our results extend to non-stochastic environments. This is the major open question that remains to be answered.

One question which we did not discuss so far is the computational efficiency of our algorithm. The issue is twofold. The first computational question is how to efficiently decide which of the four classes a given game $(\mathbf{L}, \mathbf{H})$ belongs to. The second question is the computational efficiency of BALATON for a fixed easy game. Fortunately, in both cases an efficient implementation is possible, *i.e.,* in polynomial time by using a linear program solver (*e.g.,* the ellipsoid method (Papadimitriou and Steiglitz, 1998)).

Another interesting open question is to investigate the dependence of regret on quantities other than T such as the number of actions, the number of outcomes, and more generally the structure of the loss and feedback matrices.

Finally, let us note that our results can be extended to a more general framework, similar to that of Pallavi et al. (2011), in which a game with N actions and M-dimensional outcome space is defined as a tuple $\mathbf{G} = (\mathbf{L}, S_1, \ldots, S_N)$. The loss matrix is $\mathbf{L} \in \mathbb{R}^{N\times M}$ as before, but the outcome and the feedback are defined differently. The outcome y is an arbitrary vector from a bounded subset of $\mathbb{R}^M$ and the feedback received by the learner upon choosing action i is $O_i = S_i y$.

References

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008)*, pages 263–273. Citeseer, 2008.

Alekh Agarwal, Peter Bartlett, and Max Dama. Optimal allocation strategies for the dark pool problem. In *13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010), May 12-15, 2010, Chia Laguna Resort, Sardinia, Italy*, 2010.

András Antos, Gábor Bartók, Dávid Pál, and Csaba Szepesvári. Toward a classification of finite partial-monitoring games, 2011. `http://arxiv.org/abs/1102.2041`.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Gábor Bartók, Dávid Pál, and Csaba Szepesvári. Toward a Classification of Finite Partial-Monitoring Games. In *Proceedings of the 21st international conference on Algorithmic Learning Theory (ALT 2010)*, pages 224–238. Springer, 2010.

Nicolò Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162, June 2005.

Nicoló Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Regret minimization under partial monitoring. *Mathematics of Operations Research*, 31(3):562–580, 2006.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, New York, second edition, 2006.

Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the 16th annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, page 394. Society for Industrial and Applied Mathematics, 2005.

Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science 2003 (FOCS 2003)*, pages 594–605. IEEE, 2003.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

Gábor Lugosi and Nicolò Cesa-Bianchi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

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 W. W. Cohen, A. McCallum, and S. T. Roweis, editors, *ICML 2008*, pages 672–679. ACM, 2008.

A. Pallavi, R. Zheng, and Cs. Szepesvári. Sequential learning for optimal monitoring of multi-channel wireless networks. In *INFOCOMM*, 2011.

Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Dover Publications, New York, 1998.

Antonio Piccolboni and Christian Schindelhauer. Discrete prediction games with arbitrary feedback and loss. In *Proceedings of the 14th Annual Conference on Computational Learning Theory (COLT 2001)*, pages 208–223. Springer-Verlag, 2001.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of Twentieth International Conference on Machine Learning (ICML 2003)*, 2003.

Appendix

Proof: (Lemma 8)

1. In an elimination set, we eliminate every action whose cell is contained in a closed half space. Let us assume that $j \in A_i$ is being eliminated. According to the definition of A_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 $\varepsilon > 0$, $p' \in C_k \in A_i$, and at the same time, $p' \notin C_i$. Thus we have

$$\begin{aligned} \ell_k^\top \left(p + \varepsilon \left(p - p^*\right)\right) &\leq \ell_i^\top \left(p + \varepsilon \left(p - p^*\right)\right) \\ (1+\varepsilon)\ell_k^\top p - \varepsilon \ell_k^\top p^* &\leq (1+\varepsilon)\ell_i^\top p - \varepsilon \ell_i^\top p^* \\ -\varepsilon \ell_k^\top p^* &\leq -\varepsilon \ell_i^\top p^* \\ \ell_k^\top p^* &\geq \ell_i^\top p^* \\ \alpha_{k,j^*} &\geq \alpha_{i,j^*} \,, \end{aligned}$$

   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 $\alpha_{r,s}$ is continuous in p, the limit has the desired property as well.

■

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 11 *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 $\mathbb{V}ar[X_t|\mathcal{F}_{t-1}] \leq \sigma^2$ a.s., where R, $\mu \neq 0$ and σ^2 are non-random constants. Let $p > 1$, $\epsilon > 0$, $0 < \delta < 1$ and let*

$$L_n = (1+\varepsilon) \max_{1 \leq t \leq n} \left\{ |\overline{X}_t| - c_t \right\}, \quad \textit{and} \quad U_n = (1-\varepsilon) \min_{1 \leq t \leq n} \left\{ |\overline{X}_t| + c_t \right\} ,$$

where $c_t = c(\sigma, R, t, \delta)$, and $c(\cdot)$ is defined in (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 $\tau = \min\{n \,:\, L_n \geq U_n\}$. Then, with probability at least $1 - \delta$,

$$|\hat{\mu}_\tau - \mu| \leq \varepsilon \, |\mu| \qquad \textit{and} \qquad \tau \leq C \cdot \max\left(\frac{\sigma^2}{\epsilon^2 \mu^2}, \frac{R}{\epsilon|\mu|} \right) \left(\log \frac{1}{\delta} + \log \frac{R}{\epsilon|\mu|} \right) ,$$

where $C > 0$ is a universal constant.

Lemma 12 *Fix a probability vector $p \in \Delta_M$, and let $\epsilon \in \mathbb{R}^M$ such that $p - \epsilon, p + \epsilon \in \Delta_M$ also holds. Then*

$$\mathrm{KL}(p - \epsilon || p + \epsilon) = O(\|\epsilon\|_2^2) \qquad \textit{as } \epsilon \to 0.$$

The constant and the threshold in the $O(\cdot)$ notation depends on p.

Proof: Since p, $p+\epsilon$, and $p-\epsilon$ are all probability vectors, notice that $|\epsilon(i)| \leq p(i)$ for $1 \leq i \leq M$. So 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. Since we are interested only in the case when $\epsilon \to 0$, we can also assume without loss of generality that $|\epsilon(i)| \leq p(i)/2$. Also note that the coordinates of $\epsilon = (p+\epsilon) - \epsilon$ have to sum up to zero. By definition,

$$\mathrm{KL}(p-\epsilon||p+\epsilon) = \sum_{i=1}^{M} (p(i) - \epsilon(i)) \log \frac{p(i)-\epsilon(i)}{p(i)+\epsilon(i)}.$$

We write the term with the logarithm

$$\log \frac{p(i)-\epsilon(i)}{p(i)+\epsilon(i)} = \log\left(1 - \frac{\epsilon(i)}{p(i)}\right) - \log\left(1 + \frac{\epsilon(i)}{p(i)}\right),$$

so that we can use that, by second order Taylor expansion around 0, $\log(1-x) - \log(1+x) = -2x + r(x)$, where $|r(x)| \leq c|x|^3$ for $|x| \leq 1/2$ and some $c > 0$. Combining these equations, we get

$$\begin{aligned}\mathrm{KL}(p-\epsilon||p+\epsilon) &= \sum_{i=1}^{M} (p(i)-\epsilon(i)) \left[-2\frac{\epsilon(i)}{p(i)} + r\left(\frac{\epsilon(i)}{p(i)}\right)\right] \\ &= \sum_{i=1}^{M} -2\epsilon(i) + \sum_{i=1}^{M} 2\frac{\epsilon^2(i)}{p(i)} + \sum_{i=1}^{M} (p(i)-\epsilon(i)) r\left(\frac{\epsilon(i)}{p(i)}\right).\end{aligned}$$

Here the first term is 0, letting $\underline{p} = \min_{i \in \{1,\ldots,M\}} p(i)$ the second term is bounded by $2\sum_{i=1}^{M} \epsilon^2(i)/\underline{p} = (2/\underline{p})\|\epsilon\|_2^2$, and the third term is bounded by

$$\begin{aligned}\sum_{i=1}^{M} (p(i)-\epsilon(i)) \left| r\left(\frac{\epsilon(i)}{p(i)}\right)\right| &\leq c \sum_{i=1}^{M} \frac{p(i)-\epsilon(i)}{p^3(i)} |\epsilon(i)|^3 \\ &\leq c \sum_{i=1}^{M} \frac{|\epsilon(i)|}{p^2(i)} \epsilon^2(i) \\ &\leq \frac{c}{2} \sum_{i=1}^{M} \frac{1}{\underline{p}} \epsilon^2(i) = \frac{c}{2\underline{p}} \|\epsilon\|_2^2.\end{aligned}$$

Hence, $\mathrm{KL}(p-\epsilon||p+\epsilon) \leq \frac{4+c}{2\underline{p}} \|\epsilon\|_2^2 = \mathcal{O}(\|\epsilon\|_2^2)$. ■