Toward a Classification of Finite Partial-Monitoring Games

Gábor Bartók (***student***), Dávid Pál, and Csaba Szepesvári

Department of Computing Science, University of Alberta, Canada
{bartok,dpal,szepesva}@cs.ualberta.ca

Abstract. In a finite partial-monitoring game against Nature, the Learner repeatedly chooses one of finitely many actions, the Nature responds with one of finitely many outcomes, the Learner suffers a loss and receives feedback signal, both of which are fixed functions of the action and the outcome. The goal of the Learner is to minimize its total cumulative loss. We make progress towards classification of these games based on their minimax expected regret. Namely, we classify almost all games with two outcomes: We show that their minimax expected regret is either zero, $\widetilde{\Theta}(\sqrt{T})$, $\Theta(T^{2/3})$, or $\Theta(T)$ and we give a simple and efficiently computable classification of these four classes of games. Our hope is that the result can serve as a stepping stone toward classifying all finite partial-monitoring games.

1 Introduction

A full information matrix game is specified by a finite loss matrix, $L = (\ell_{ij})$, where $1 \le i \le N$ denotes the actions of the row player and $1 \le j \le M$ denotes the actions of the column player, while $\ell_{ij} \in [0,1]$ is the loss suffered by the row player when he chose action i and the opponent chose action j. In games against Nature, Nature plays the role of the column player. In these games, at the beginning of the game Nature chooses an arbitrary sequence of actions of length T, unknown to the row player (henceforth Learner). If the sequence was known, the Learner could select the action that gives rise to the smallest possible cumulated loss. The regret of the Learner is defined by his excess cumulated loss compared to the mentioned best possible cumulated loss. Generally, the regret grows with the horizon. If the growth is sublinear then in the long run the Learner can be said to play almost as well as if he knew Nature's sequence of actions in advance. In a *full information matrix game against Nature*, the Learner is told Nature's action after every round, so that he has a chance to make adjustments to what actions to play. The Learner in general needs to randomize to prevent being second-guessed. In this situation, it is known that the Learner can keep his expected regret, R_T, below $\sqrt{T \ln(N)/2}$, independently of M (cf. Chapter 4 and the references in the book by Lugosi and Cesa-Bianchi [1]).

When playing in a *partial-information matrix game*, the main topic of this article, Nature's actions can be masked. More precisely, at the beginning of the game the Learner is given a pair of $N \times M$ matrices, (L, H), where L is the loss matrix as before, while H is a matrix that specifies what information the Learner receives in the rounds. The elements of H belong to some alphabet, which, without the loss generality (WLOG), can be assumed to be the set of natural numbers. The way the game is played is modified as follows: in round t if i and j are the actions chosen by the Learner and Nature, respectively, then instead of j, the Learner receives H_{ij} only as the feedback. It is then the structure of H that determines how much information is revealed in each time step: assuming the learner selects i, H_{ij} may reveal the identity of j (i.e., if $H_{ij} \neq H_{ik}$, $1 \leq j < k \leq M$) or it may mask it completely (i.e., if $H_{i,\cdot} \equiv \text{const}$). The goal of the Learner is still to keep its regret small, but the game has now a new element: The learner might need to give up on using actions with small losses in favour of playing informative actions i.e., the *exploration vs. exploitation tradeoff* appears.

Let us now discuss some previous results, followed by a short description of our contributions. A special case of partial-information games is when the Learner learns the loss of the action taken (i.e., when $H = L$), also known as the *bandit case.* Then, the INF algorithm due to Audibert and Bubeck [2] is known to achieve a regret bound $O(\sqrt{NT})$. (The Exp3 algorithm due to Auer et al. [3] achieves the same bound up to logarithmic factors.) It is also known that this is the best possible bound [3].

Now, consider another special case: Imagine a 3×2 game (L, H), where the first action of the Learner gives full information about Nature's choice ($H_{11} \neq H_{12}$), but it has a high cost, independently of Nature's choice (say, $\ell_{11} = \ell_{12} = 1$), while the other two actions do not reveal any information about Nature's choice (i.e., $H_{i1} = H_{i2}$, $i = 2, 3$). Further, assume that the cost of action 2 is low if Nature chooses action 1 and the cost of action 3 is low if Nature chooses action 2, say, $\ell_{21} = \ell_{32} = 0$, $\ell_{22} = \ell_{31} = 1$:

$$L = \begin{pmatrix} 1\,1 \\ 0\,1 \\ 1\,0 \end{pmatrix}, \qquad H = \begin{pmatrix} 0\,1 \\ 0\,0 \\ 0\,0 \end{pmatrix}.$$

In this case, it is known that the regret growth-rate is bounded by $\Omega(T^{2/3})$ from below (cf. Theorem 3, [4]), showing that a game like this is intrinsically harder than a bandit problem. Further, it is known that the regret of the "general forecaster for partial monitoring" is bounded by $O(T^{2/3})$ (cf. Theorem 3.1, [4]).

It is also clear that in certain cases the best possible regret grows linearly with T (i.e., when no information is received about Nature's actions), while in some other trivial cases the learner can achieve 0 regret.

Thus, we see, that the difficulty of a game depends on the structure of L and H. However, as of yet, it is unclear what determines this difficulty. In fact, discussing the rate of decay of the regret per time-step, Cesa-Bianchi et al. [4] note that *"It remains a challenging problem to characterize the class of problems that admit rates of convergence faster than $O(T^{-1/3})$"*.[1] This is exactly the question

[1] Here we renamed their n to T to match our notation.

which motivated the research reported on in this paper. In particular, we wish to answer the following questions:

1. Given L, H, how difficult is the game $G = (L, H)$? That is, given G what is the growth-rate of *the minimax regret*,
$$\mathrm{R}_T(G) = \inf_{A\in\mathcal{A}} \sup_{E\in\mathcal{E}} \mathrm{R}_T(G, A, E), \tag{1}$$
corresponding to G, where $\mathcal{A}$ is the class of randomized strategies for the learner, $\mathcal{E}$ is the class of Nature's strategies and $\mathrm{R}_T(G, A, E)$ denotes the expected regret up to time T when the Learner using strategy A is playing against Nature in game G and Nature uses strategy E.
2. Do there exist games where the exponent of the minimax regret rate is other than 0, 1/2, 2/3, and 1?
3. Does there exist a strategy (and what is it), which, when fed with $G = (L, H)$, achieves the minimax regret?

In this paper, we make some initial steps toward answering these questions. In particular, for games *when Nature has at most two actions*, apart from a set of games of measure zero, we give complete answer to the above questions.

In particular, we show that the answer to the second question above is negative: Only exponents $0, 1/2, 2/3$ and 1 can appear in the growth rate of the minimax regret. As far as the lower bounds are concerned, an exponent of 1/2 follows since partial-monitoring games are clearly at least as difficult as full-information games and, for the latter games, as it is well known, a lower bound on the minimax regret with exponent 1/2 holds [5]. Thus, our first contribution is to show that if the exponent of the minimax regret rate is above 1/2 then it cannot be below 2/3. Precisely, we show that for the said set of games if in the chain of nondominated actions of a game there exists two consecutive actions under which Nature's two actions are indistinguishable (i.e., the actions are non-revealing) then Nature can force a high regret. Here, an action i is called *nondominated* if there exists a distribution over Nature's actions such that action i has the smallest average loss over that distribution. Otherwise it is called *dominated.* An action i is *non-revealing* if $H_{i1} = H_{i2}$, otherwise it is called *revealing.*

Our next contribution is that we give a strategy which, apart from these difficult games, achieves a regret growth rate with exponent 1/2. Here, the insight is that if at least one of any pair of consecutive nondominated actions is a revealing action then the Learner can gain enough information cheaply. Since Corollary 4.2 due to Cesa-Bianchi et al. [4] states that a regret with exponent 2/3 is achievable for all non-trivial partial-monitoring games, we basically get a complete classification of games with $M = 2$.

2 Results

Consider a finite partial-monitoring game $G = (L, H)$ of size $N \times M$. The *regret* of the Learner playing sequence $1 \le I_t \le N$ against Nature who is playing

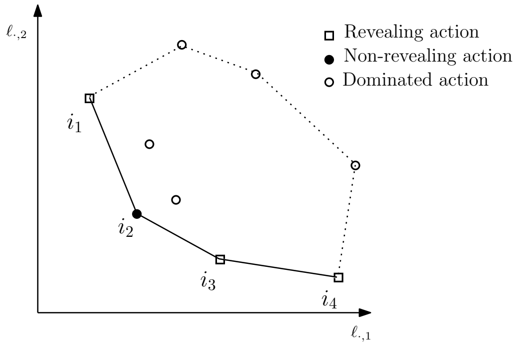

Fig. 1. The figure shows each action i as a point in $\mathbb{R}^2$ with coordinates $(\ell_{i,1}, \ell_{i,2})$. The solid line connects the chain of nondominated actions, which, by convention are ordered according to their loss for the first outcome.

$1 \le J_t \le M$ is defined as

$$\widehat{\mathrm{R}}_T = \sum_{t=1}^{T} \ell_{I_t, J_t} - \min_{1 \le i \le M} \sum_{t=1}^{T} \ell_{i, J_t} \ . \tag{2}$$

The expected regret is defined by $\mathrm{R}_T = \mathbb{E}[\widehat{\mathrm{R}}_T]$. (Note that since the Learner can randomize, $\widehat{\mathrm{R}}_T$ is a random variable.) In what follows Nature's actions will also be called *outcomes*.

From now on we consider only the case when $M = 2$. Dominated, nondominated, revealing and non-revealing actions were introduced in the introduction. These concepts can be visualized by showing each action i as a point in $\mathbb{R}^2$ with coordinates $(\ell_{i,1}, \ell_{i,2})$. Then the points corresponding to the nondominated actions lie on the boundary of the convex hull of the set of all the actions. See Figure 1. Enumerating the nondominated actions in the counter-clockwise order along the boundary of the convex hull gives rise to a sequence $(i_1, i_2, \ldots, i_K)$, which we call the *chain of nondominated actions*.

To avoid trivialities, WLOG we will assume that there are no *duplicate* actions, that is, two actions i, j, $i \neq j$, such that the i-th and the j-th rows of the loss matrix are the same. Clearly, if duplicate actions exists then at least one of them can be removed without changing the min-max expected regret: If both are either revealing or non-revealing, it does not matter which action is removed. Otherwise, we remove the non-revealing action.

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

Separation Condition. *A game G satisfies the* separation condition *if its chain of nondominated actions does* **not** *have a pair of consecutive actions i_j, i_{j+1} such that both of them are non-revealing. The set of games satisfying this condition will be denoted by $\mathcal{S}$.*

Non-degeneracy Condition. *A game G satisfies the* non-degeneracy condition *if each of its nondominated actions is an extreme*[2] *point of the convex hull of all the actions.*

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 is merely a technical condition that we need in our proofs. The Lebesgue measure of the class of loss matrices it excludes is zero. We are now ready to state our main result.

Theorem 1 (Classification of two-outcome partial-monitoring games). *Let $G = (L, H)$ be a finite partial-monitoring game with two outcomes that satisfies the non-degeneracy condition. Let $(i_1, i_2, \dots, i_K)$ be the chain of nondominated actions in G. Let $\mathcal{S}$ be the set of games satisfying the separation condition. The min-max expected regret $\mathrm{R}_T(G)$ satisfies*[3]

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

Cases (3a) and (3d) are trivial. The lower bound of case (3b) follows from the fact that even if we assume full information, the expected regret is $\Omega(\sqrt{T})$ [5]. The upper bound of case (3c) can be derived from a result of Cesa-Bianchi et al. [4]: Recall that the entries of 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 et al. [4] show that if the entries of H can be chosen such that $\mathrm{rank}(H) = \mathrm{rank}\begin{pmatrix} H \\ L \end{pmatrix}$ then $O(T^{2/3})$ expected regret is achievable. This condition holds trivially for two-outcome games with at least one revealing action. It remains to prove the upper bound for (3b) and the lower bound for (3c). We prove these in the next sections.

3 Upper bound

In this section we present our algorithm, AppleTree, for games satisfying the separation condition and the non-degeneracy condition, and prove that it achieves $\widetilde{O}(\sqrt{T})$ regret with high probability. (The choice of the name of the algorithm will be explained later.)

[2] An extreme point of a convex set is a point which is not a non-trivial convex combination of two different points of the set. In our case, the set is a convex polygon and its extreme points are precisely its vertices.

[3] Here, $a_n = \widetilde{\Theta}(b_n)$ stands for $a_n = \Omega(b_n)$ and $a_n = \widetilde{O}(b_n)$.

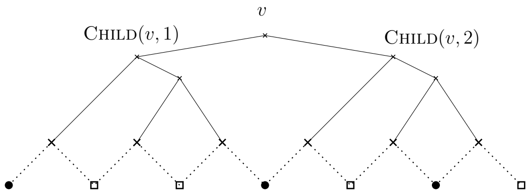

Fig. 2. The binary tree built by the algorithm. The leaf nodes represent neighboring action pairs.

3.1 Algorithm

In the first step of the algorithm we can purify 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 action revealing the outcome, the other revealing no information).

To see why this latter case corresponds to one-armed bandits assume WLOG 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] By subtracting $\ell_{2,1}$ from the first and $\ell_{2,2}$ from the second column we thus get the equivalent game where the second row of the loss matrix is zero. In this game, the Learner knows the loss of the second action independently of the outcome, while, since the first action is revealing, he learns the loss of the first action in any round when that action is played, which is exactly what one has in a one-armed bandit game. Since a one-armed bandit is a special form of a two-armed bandit, one can use Exp3.P due to Auer et al. [3] to achieve the $\widetilde{O}(\sqrt{T})$ regret[5].

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*. Here the actions are ordered according to their losses corresponding to the *first* outcome. This is continued until the split results into games with two actions only. The recursive splitting of the game results in a binary tree (see Figure 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 G_1, the first subgame (simply because then the frequency of outcome 1 is high and G_1 contains the actions with the

[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., the algorithm's regret when measured in the original, unmodified game). Piccolboni and Schindelhauer [6] exploit this idea, too.

[5] Apparently, this is a new result for this kind of game, also known as apple tasting.

smallest loss for the first outcome). Conversely, if this frequency is large, the optimal action is one of the actions of 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 $(1-\rho)\ell_{1,s-1} + \rho\ell_{2,s-1} = (1-\rho)\ell_{1,s}+\rho\ell_{2,s}$ and ρ_2^* is the solution to $(1-\rho)\ell_{1,s+1}+\rho\ell_{2,s+1} = (1-\rho)\ell_{1,s}+\rho\ell_{2,s}$, 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 G_1 or G_2, depending on whether $\rho_T \leq \rho_1^*$ or $\rho_T \geq \rho_2^*$. (When $\rho_1^* \leq \rho_T \leq \rho_2^*$ 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 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 simplicity, assume that ρ_t was available. Then one idea would be to play a strategy restricted to the actions of either game G_1 or 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 G_1 then we have to switch to the game G_2. In this case, we start a fresh copy of a strategy playing in G_2. The same happens when a switch from G_2 to game G_1 is necessary. The 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 G_1 and 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 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 3).

Let us now turn to how ρ_t is estimated. In any round, the algorithm receives feedback $h_t \in \{1, 2, *\}$: if a revealing action is played in the round, $h_t = J_t \in \{1, 2\}$, otherwise $h_t = *$. If the algorithm choosing the actions decides with probability $p_t \in (0, 1]$ to play a revealing action (p_t can depend on the history $\mathcal{H}_t$) then $\mathbb{I}(h_t = 2)/p_t$ is a simple unbiased estimate of $\mathbb{I}(J_t = 2)$ (in fact, $\mathbb{E}\left[\mathbb{I}(h_t = 2)/p_t | \mathcal{H}_t\right] = \mathbb{I}(J_t = 2)$). As long as p_t does not drop to a too low value, $\hat{\rho}_t = \frac{1}{t}\sum_{s=1}^{t} \frac{\mathbb{I}(h_t=2)}{p_t}$ will be a relatively reliable estimate of ρ_t (see Lemma 4). However reliable this estimate is, it can still differ from ρ_t. For this reason, we

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

Fig. 3. The main entry point of the AppleTree algorithm

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

Fig. 4. The initialization routine InitEta.

```
function BuildTree($v, G, \delta$)
Input: $G = (L, H)$ is a game, $v$ is a tree node
if NumberOfActions($G$) $= 2$ then
    if not IsRevealingAction($G, 1$) then
        $G \leftarrow$ SwapActions($G$)
    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($G, T$)
else
    $(G_1, G_2) \leftarrow$ SplitGame($G$)
    BuildTree(Child($v, 1$), $G_1, \delta/(4T)$ )
    BuildTree(Child($v, 2$), $G_2, \delta/(4T)$ )
    $g(v) \leftarrow 1,\ \hat{\rho}(v) \leftarrow 0,\ t(v) \leftarrow 1$
    $(\rho'_1(v), \rho'_2(v)) \leftarrow$ Boundaries($G$)
end if
$G(v) \leftarrow G$
```

Fig. 5. The tree building procedure

push the boundaries determining game switches towards each other:

$$\rho'_1 = \frac{2\rho^*_1 + \rho^*_2}{3}, \quad \rho'_2 = \frac{\rho^*_1 + 2\rho^*_2}{3}. \tag{4}$$

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 2.3 of [4]). The algorithm's main entry point is shown on Figure 3. Its inputs are the game $G = (L, H)$, the 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, which is used to store variables necessary to play in the subgames (Figure 5): If the number of actions is 2, the procedure initializes various parameters that are used either by a bandit algorithm (based on Exp3.P [3]), or by the exponentially weighted average algorithm (EWA) [5]. In the other case, it calls itself recursively on the splitted subgames and with an appropriately decreased confidence parameter.

The main worker routine is called Play. This is again a recursive function (see Figure 6). 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,

```
function PLAY($v$)
Input: $v$ is a tree node
if ACTIONNUMBER($G(v)$) = 2 then
    $(p,h) \leftarrow$ PLAYLEAF($v$)
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 $(p,h)$
```

Fig. 6. The recursive function PLAY

```
function RESET($v$)
Input: $v$ is a tree node
if ACTIONNUMBER($G(v)$) = 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
```

Fig. 7. Function RESET

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 Figure 7, is also recursive.

At the leaves, when there are only two actions, either EWA or Exp3.P is used. These algorithms are used with their standard optimized parameters (see Corollary 4.2 for the tuning of EWA, and Theorem 6.10 for the tuning of Exp3.P, both from the book of Lugosi and Cesa-Bianchi [1]). For completeness, their pseudocodes are shown in Figures 8–9. Note that with Exp3.P (lines 6–14) we use the loss matrix transformation described earlier, hence the loss matrix has zero entries for the second (non-revealing) action, while the entry for action 1 and outcome j is $\ell_{1,j}(v) - \ell_{2,j}(v)$. Here $\ell_{i,j}(v)$ stands for the loss of action i and outcome j in the game $G(v)$ that is stored at node v.

3.2 Proof of the upper bound

Theorem 2. *Assume $G = (L,H)$ satisfies the separation condition and the non-degeneracy condition and $\ell_{i,j} \leq 1$. Denote by $\widehat{\mathrm{R}}_T$ the regret of Algorithm* APPLE-TREE *up to time step T. There exist constants c, p such that for any $0 < \delta < 1$ and $T \in \mathbb{N}$, the algorithm with input G, T, δ 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. To any variable $x(v)$ used by the algorithm, we denote by $x_t(v)$ 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

```
function PlayAtLeaf($v$)
Input: $v$ is a tree node
if RevealingActionNumber($G(v)$) $= 2$ then        ▷ Full information case
    $(p, h) \leftarrow$ Ewa($v$)
else                                               ▷ Partial information case
    $p \leftarrow (1 - \gamma(v)) \frac{w_1(v)}{w_1(v)+w_2(v)} + \gamma(v)/2$
    $U \sim \mathcal{U}_{[0,1)}$                   ▷ $U$ is uniform in $[0, 1)$
    if $U < p$ then                                ▷ Play revealing action
        $h \leftarrow$ PLAY(1)                     ▷ $h \in \{1, 2\}$
        $L_1 \leftarrow (\ell_{1,h}(v) - \ell_{2,h}(v) + \beta(v))/p$
        $L_2 \leftarrow \beta(v)/(1 - p)$
        $w_1(v) \leftarrow w_1(v) \exp(-\eta(v) L_1)$
        $w_2(v) \leftarrow w_2(v) \exp(-\eta(v) L_2)$
    else
        $h \leftarrow$ PLAY(2)                     ▷ here $h = *$
    end if
end if
return $(p, h)$
```

Fig. 8. Function PlayAtLeaf

```
function Ewa($v$)
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
$h \leftarrow$ PLAY($I$)                           ▷ $h \in \{1, 2\}$
$w_1(v) \leftarrow w_1(v) \exp(-\eta(v) \ell_{1,h}(v))$
$w_2(v) \leftarrow w_2(v) \exp(-\eta(v) \ell_{2,h}(v))$
return $(p, h)$
```

Fig. 9. Function Ewa

now on we 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 3. *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 $\Delta = \rho_2' - \rho_1'$, ρ_1' and ρ_2' given by* (4).

Note that here we use the non-degeneracy condition to ensure that $\Delta > 0$. *Proof* Let s be the number of times the algorithm switches from G_2 to G_1. Let $t_1 < \ldots < t_s$ be the time steps when $\hat{\rho}_t$ becomes smaller than ρ_1'. Similarly, let $t_1' < \ldots < t_{s+\xi}'$, $(\xi \in \{0, 1\})$ be the time steps when $\hat{\rho}_t$ becomes greater than ρ_2'. Note that for all $1 \leq j < s$, $t_j' < t_j < t_{j+1}'$. The number of times the algorithm resets is at most $2s + 1$. For any $1 \leq j \leq s$, $\hat{\rho}_{t_j'} > \rho_2'$ and $\hat{\rho}_{t_j} < \rho_1'$. According to the update rule we have for any t that

$$\hat{\rho}_t = \left(1 - \frac{1}{t}\right) \hat{\rho}_{t-1} + \frac{1}{t} \cdot \frac{\mathbb{I}(J_t = 2)}{p_t} \geq \frac{t-1}{t} \hat{\rho}_{t-1} = \hat{\rho}_{t-1} - \frac{1}{t} \hat{\rho}_{t-1}$$

and hence $\hat{\rho}_{t-1} - \hat{\rho}_t \leq \frac{1}{t}$. Summing this inequality for all $t_j' + 1 \leq t \leq t_j$ we get $\Delta \leq \hat{\rho}_{t_j'} - \hat{\rho}_{t_j} \leq \sum_{t=t_j'}^{t_j - 1} \frac{1}{t} = O\left(\ln \frac{t_j}{t_j'}\right)$, using that $\Delta = \rho_2' - \rho_1'$. Thus, there exists $c^* > 0$ such that for all $1 < j \leq s$

$$\frac{1}{c^*} \Delta \leq \ln \frac{t_j}{t_j'} \leq \ln \frac{t_j}{t_{j-1}} \ . \tag{5}$$

Adding (5) for $1 < j \le s$ we get $(s-1)\frac{1}{c^*}\Delta \le \ln \frac{t_s}{t_1} \le \ln T$, which yields the desired statement. □

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

Lemma 4. *Let $c = \frac{8}{3\Delta^2}$. Then for any $0 < \delta < 1$, with probability at least $1-\delta$, for all $t \ge c\sqrt{T}\ln(2T/\delta)$, $|\hat{\rho}_t - \rho_t| \le \Delta$.*

Proof Using *Bernstein's inequality for martingales* (see Lemma A.8 in Lugosi and Cesa-Bianchi [1]) and the fact that, due to the construction of the algorithm, the probability p_t of playing a revealing action at time step t is always greater than $1/\sqrt{T}$, we get that for any t $\Pr\left(|\hat{\rho}_t - \rho_t| > \Delta\right) \le 2\exp\left(-\frac{3\Delta^2 t}{8\sqrt{T}}\right)$. Reordering the inequality and applying the union bound for all $1 \le t \le T$ we get the result. □

Proof of Theorem 2 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 EWA or Exp3.P. EWA is known to satisfy Theorem 2, 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 2 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 set of actions associated with the subgames in the root[6]. Furthermore, let us define the following values: Let $T_0^0 = 1$, let T_i^0 be the first time step after T_{i-1}^0 such that $g_t \ne g_{t-1}$. In other words, T_i^0 are the time steps when the algorithm switches between the subgames. Finally, let $T_i = \min(T_i^0, T)$. From Lemma 3 we know that $T_{S_{max}+1} = T$, where $S_{max} = \frac{c^* \ln T}{\Delta}$. It is easy to see that T_i are stopping times for any $i \ge 1$.

WLOG, from now on we will assume that the optimal action is action 1. Let $S = \arg\max\{i \ge 1 | T_i^0 \le T\}$ the number of switches and $\mathcal{B}$ be the event that for all $t \ge c\sqrt{T}\ln(4T/\delta)$, $|\hat{\rho}_t - \rho_t| \le \Delta$. We know from Lemma 4 that $\mathbb{P}(\mathcal{B}) \ge 1 - \delta/2$. On $\mathcal{B}$ we have that $|\hat{\rho}_T - \rho_T| \le \Delta$, and thus, using that action 1 is optimal, $\rho_T \le \rho_1^*$. This implies that in the last phase the algorithm plays on 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 $\sum_{t=1}^{T_S-1} \ell_{1,J_t} - \ell_{\lceil K/2 \rceil, J_t} \ge 0$. On the other hand, if $T_S \le c\sqrt{T}\ln(4T/\delta)$ then

$$\sum_{t=1}^{T_S-1} \ell_{1,J_t} - \ell_{\lceil K/2 \rceil, J_t} \ge -c\sqrt{T}\ln(4T/\delta) \, .$$

[6] Recall that the actions are ordered with respect to $\ell_{\cdot,1}$.

Thus, we have

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

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 makes 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 $\widehat{\mathrm{R}}_m^{(r)} = \sum_{t=T_{r-1}}^{T_{r-1}+m-1} (\ell_{I_t,J_t} - \ell_{i,J_t})$ is the regret of the algorithm if it is used in the subgame $G_{\pi(r)}$ 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 $\mathbb{P}(\mathcal{C}^c) \le \delta/2$. In summary,

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

Thus, on $\mathcal{B} \cap \mathcal{C}$, $\widehat{\mathrm{R}}_T \le \frac{2^p c c^*}{\varDelta}\sqrt{T}\ln^{p+1}(2T/\delta)$, which, together with $\mathbb{P}(\mathcal{B}^c \cup \mathcal{C}^c) \le \delta$ 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 $1/T$. Also note that the bound given by the induction grows in the number of nondominated actions as $O(K^{\log_2 K})$.

4 Lower Bound

In this section we present a lower bound for the expected regret in the case when the separation condition does not hold.

Theorem 5. *If the chain of nondominated actions of G satisfies the non-degeneracy condition and the separation condition does **not** hold then for any algorithm A and time horizon T there exists a sequence of outcomes such that the expected regret $\mathrm{R}_T(A)$ of the algorithm satisfies $\mathrm{R}_T(A) = \Omega(T^{2/3})$.*

Proof We follow the steps of the lower bound proof for the label efficient prediction from Cesa-Bianchi et al. [4] with a few changes. The most important change, as we will see, is the choice of the models we randomize over.

We can assume WLOG that actions 1 and 2 are the two consecutive nondominated non-revealing actions, while all the other actions are revealing and $(\ell_{1,1}, \ell_{1,2}) = (0, \alpha)$, $(\ell_{2,1}, \ell_{2,2}) = (1-\alpha, 0)$ with some $\alpha \in [0, 1]$. That this can be assumed follows by scaling and a reduction similar to the one we used in Section 3.1. Using the non-degeneracy condition and that actions 1 and 2 are consecutive, we get that for all $i \geq 3$, there exists some $\lambda_i \in \mathbb{R}$ such that

$$\begin{aligned} \ell_{i,1} &> \lambda_i \ell_{1,1} + (1-\lambda_i)\ell_{2,1} = (1-\lambda_i)(1-\alpha) \, , \\ \ell_{i,2} &> \lambda_i \ell_{1,2} + (1-\lambda_i)\ell_{2,2} = \lambda_i \alpha \, . \end{aligned} \tag{6}$$

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

We construct random outcome sequences as follows. We define two models for generating outcome sequences. We use $p_i(\cdot)$ and $\mathbb{E}_i[\cdot]$ to denote probability mass function and expectation given model $i \in \{1, 2\}$, respectively. In model 1 the outcomes are i.i.d. random variables with $p_1(1) = \alpha + \epsilon$ whereas in model 2, $p_2(1) = \alpha - \epsilon$ with $\epsilon < 1$ to be chosen later. Note that, if ϵ is small enough then only actions 1 and 2 can be optimal. Namely, action i is optimal in model i.

Let $h_t \in \{*, 1, 2\}$ denote the observation of the algorithm at time step t, and let h^t denote the observation sequence $(h_1, \ldots, h_t)$. Let $A_t(h^{t-1})$ denote the choice of the algorithm[7] at time step t, given the history of observations h^{t-1}. Let $N_i^j = \mathbb{E}_j[\sum_{t=1}^T \mathbb{I}(I_t = i)]$, that is, the expected number of times action i is played up to time step T, given model j. Finally, let $N_{\geq 3}^j = \sum_{i \geq 3} N_i^j$.

Let $D(p||q)$ be the KL divergence of Bernoulli distributions with parameters p and q. We need the following technical lemma.

Lemma 6. *Let $0 < \epsilon < \alpha$ be such that $\alpha + \epsilon < 1$. Then $D(\alpha - \epsilon || \alpha + \epsilon) = \frac{2\epsilon^2}{\alpha(1-\alpha)} + O\left(\epsilon^3\right)$.*

Proof The result follows from the definition of KL divergence and the second order Taylor expansion of $\ln(1 + x)$. □

The next lemma states that the expected number of times actions 1 and 2 are played by A does not change too much if we change the model:

Lemma 7. *There exists a constant c (depending on α only) such that*

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

[7] Conditioning on the internal randomization of A if necessary, we can assume WLOG that algorithm A is deterministic.

Proof We only prove the first inequality, the other one is symmetric. We have

$$\begin{aligned} N_2^2 - N_2^1 &= \sum_{h^{T-1}} \left(p_2\left(h^{T-1}\right) - p_1\left(h^{T-1}\right)\right) \sum_{t=1}^{T} \mathbb{I}\left(A_t\left(h^{t-1}\right) = 2\right) \\ &\leq T \sum_{h^{T-1}} \left(p_2\left(h^{T-1}\right) - p_1\left(h^{T-1}\right)\right) \\ &\leq T \sum_{\substack{h^{T-1}: \\ p_2\left(h^{T-1}\right) \geq p_1\left(h^{T-1}\right)}} \left(p_2\left(h^{T-1}\right) - p_1\left(h^{T-1}\right)\right) \\ &= \frac{T}{2} \|p_2 - p_1\|_1 \leq c_1 T \sqrt{D\left(p_2 || p_1\right)}, \end{aligned}$$

where the last step follows from Pinsker's inequality [7]. Using the chain rule for KL divergence we can write

$$\begin{aligned} D\left(p_2||p_1\right) &= \sum_{t=1}^{T} D\left(p_2(h_t|h^{t-1})||p_1(h_t|h^{t-1})\right) \\ &= \sum_{t=1}^{T} \sum_{h^{t-1}} p_2(h^{t-1}) \sum_{h_t} p_2(h_t|h^{t-1}) \ln \frac{p_2(h_t|h^{t-1})}{p_1(h_t|h^{t-1})} \\ &= \sum_{t=1}^{T} \sum_{h^{t-1}} \mathbb{I}(A_t(h^{t-1}) \geq 3) p_2(h^{t-1}) \sum_{h_t \in \{1,2\}} p_2(h_t|h^{t-1}) \ln \frac{p_2(h_t|h^{t-1})}{p_1(h_t|h^{t-1})} \qquad (7) \\ &= \sum_{t=1}^{T} \sum_{h^{t-1}} \mathbb{I}(A_t(h^{t-1}) \geq 3) p_2(h^{t-1}) \left(\frac{2\epsilon^2}{\alpha(1-\alpha)} + O\left(\epsilon^3\right)\right) \qquad (8) \\ &= \left(\frac{2\epsilon^2}{\alpha(1-\alpha)} + O\left(\epsilon^3\right)\right) N_{\geq 3}^2, \end{aligned}$$

In (7) we used that if we play action 1 or 2 then our observation h_t will be $*$ in both models 1 and 2, whereas if we play action $i \geq 3$ then $h_t \in \{1, 2\}$, while in (8) we used Lemma 6. □

The expected regret of the algorithm can be bounded in terms of N_i^j:

$$\mathbb{E}_1[\widehat{\mathrm{R}}_T] \geq (\ell_1^1(\alpha+\epsilon) + \ell_2^1(1-\alpha-\epsilon) - \alpha(1-\alpha-\epsilon))N_{\geq 3}^1 + \epsilon N_2^1 = f_1 N_{\geq 3}^1 + \epsilon N_2^1$$
$$\mathbb{E}_2[\widehat{\mathrm{R}}_T] \geq (\ell_1^2(\alpha-\epsilon) + \ell_2^2(1-\alpha+\epsilon) - (1-\alpha)(\alpha-\epsilon))N_{\geq 3}^2 + \epsilon N_1^2 = f_2 N_{\geq 3}^2 + \epsilon N_1^2$$

where, for an outcome i, ℓ_i^j is the loss of the best revealing action given model j. Now, by (6), there exists $\tau > 0$ such that for all $i \geq 3$, $\ell_{i,1} \geq (1-\lambda_i)(1-\alpha) + \tau$ and $\ell_{i,2} \geq \alpha\lambda_i + \tau$. Simple algebra gives that $f_1 \geq (1-\lambda_{max})\epsilon + \tau$ and $f_2 \geq \lambda_{min}\epsilon + \tau$. Hence, if ϵ is small enough then both f_1 and f_2 are positive. Therefore, choosing $j = \arg\min_{l \in \{1,2\}}(N_{\geq 3}^l)$ and using Lemma 7 we get $\mathbb{E}_i[\widehat{\mathrm{R}}_T] \geq f_i N_{\geq 3}^j + \epsilon\left(N_{3-i}^j - cT\epsilon\sqrt{N_{\geq 3}^j}\right)$, $i = 1, 2$. Finally, randomizing over the two models such that each of them is chosen with equal probability and denoting the corresponding expectation by $\mathbb{E}[\cdot]$, setting ϵ to $c_2 T^{-1/3}$ we have $\mathbb{E}[\widehat{\mathrm{R}}_T] \geq (\tau - \frac{\lambda^* c_2 T^{-1/3}}{2}) N^j_{\geq 3} + c_2 T^{2/3} - c_2^2 c T^{1/3} \sqrt{N^j_{\geq 3}} > T^{2/3} \left((\tau - \frac{\lambda^* c_2}{2}) x^2 + c_2 - c_2^2 c x \right)$, where $x = \sqrt{\frac{N^j_{\geq 3}}{T^{2/3}}}$. Now it is easy to see that c_2 can be set such that, independently of x, the right hand side is always positive and thus it is $\Omega(T^{2/3})$.

5 Discussion

In this paper we classified partial-monitoring games with two outcomes based on their minimax regret. The most important open question is whether our results generalize to games with more outcomes.

A simple observation is that, given a finite partial-monitoring game, if we restrict Nature's set of actions 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. We believe that the $\Omega(T^{2/3})$ lower bound can also be generalized to situations where two "ϵ-close" outcome distributions are not distinguishable by playing only their respective optimal actions. Generalizing the upper bound result seems more challenging. The algorithm AppleTree heavily exploits the two-dimensional structure of the losses and, as of yet, in general we do not know how to construct an algorithm that achieves $\widetilde{O}(\sqrt{T})$ regret on partial-monitoring games with more than two outcomes.

Bibliography

[1] G. Lugosi and N. Cesa-Bianchi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

[2] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

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

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

[5] N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3): 427–485, 1997.

[6] 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)*, pages 208–223. Springer-Verlag, 2001.

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