SIIM-TR-A-00-18

# SIIM Technical Report

# Discrete Prediction Games with Arbitrary Feedback and Loss

**by**

ANTONIO PICCOLBONI, CHRISTIAN SCHINDELHAUER

**Schriftenreihe der Institute für**

**Informatik/Mathematik**

**Serie A**

November 3, 2000

Medizinische
Universität zu Lübeck
Technisch-Naturwissenschaftliche Fakultät

Email: schindel@informatik.mu-luebeck.de Phone: +49-451-7030-414

Fax: +49-451-7030-438

# Discrete Prediction Games with Arbitrary Feedback and Loss

Antonio Piccolboni* Christian Schindelhauer†

November 3, 2000‡

**Abstract**

We investigate the problem of predicting a sequence when the information about the previous elements (feedback) is only partial and possibly dependent on the predicted values. This setting can be seen as a generalization of the classical multi-armed bandit problem and accommodates as a special case a natural bandwidth allocation problem. According to the approach adopted by many authors, we give up any statistical assumption on the sequence to be predicted. We evaluate the performance against the best constant predictor (regret), as it is common in iterated game analysis.

We show that for any discrete loss function and feedback function only one of two situations can occur: either there is a prediction strategy that achieves in $T$ rounds a regret of at most $O(T^{3/4}(\ln T)^{1/2})$ or there is a sequence which cannot be predicted by any algorithm without incurring a regret of $\Omega(T)$.

We prove both sides constructively, that is when the loss and feedback functions satisfy a certain condition, we present an algorithm that generates predictions with the claimed performance; otherwise we show a sequence that no algorithm can predict without incurring a linear regret with probability at least $1/2$.

## 1 Introduction

Our research was initially prompted by the following bandwidth allocation problem. On the link between two servers, a varying bandwidth is available. As it is common in an internetworking setting, little or no information is available about load patterns for the link and no cooperative behavior can be guaranteed. The goal is to send data

*International Computer Science Institute, Berkeley. Email: piccolbo@icsi.berkeley.edu.

†Institut für Theoretische Informatik, Med. Universität zu Lübeck. Email: schindel@tcs.mu-luebeck.de.
Parts of this work are supported by a stipend of the "Gemeinsames Hochschulsonderprogramm III von Bund und Länder" through the DAAD.

‡Revised Version (February 1, 2001): Lemma 1 is inserted and Lemma 3, Lemma 6 and the proof of Theorem 3 are rewritten.

as quickly as possible without exceeding the bandwidth available, assuming that some price is paid in case of congestion. Due to the limitations of typical protocols, the only feedback servers receive is whether congestion occurred or not (in the form, e.g., of a dropped packet). An algorithm choosing the bandwidth is fronting a trade off similar to the one that is the most distinctive trait of the multi-armed bandit problem: on one hand, trying to match the maximum bandwidth at any time step; on the other, choosing the bandwidth in order to collect more information about the load.

Another, even simpler, instance of this general setting arises from a simple quality control problem. In a manufacturing operation, the items produced can be either working or defective. Unfortunately, to asses the quality of an item it is necessary to destroy it. Both delivering a defective item and destroying a working one are undesirable events. Suppose that customer feedback is unavailable, late or unreliable. The only information available about the sequence of items produced so far is the one the destructive testing procedure provides, but we want to apply it as little as possible. When the plant is working properly, defective items are extremely rare so that little testing seems optimal, but a failure would be detected with a worrisome delay.

The goal we set for ourselves was to make these two examples, together with the multi-armed bandit problem and others, fit a general framework that encompasses different sequence prediction games where the prediction is based only on some "clues" about the past rounds of the game and good predictions are rewarded according to some weighting scheme. We model the available feedback on the sequence as a function of two arguments. One is the current sequence value itself, as it is common in system theory, and the other is the prediction itself. In system theory the classic problem is that of observability: is the feedback sufficient to find out the initial state of the system, whose transition function is assumed to be known? More closely related to our problem is that of learning from noisy observations, where the sequence is obfuscated by some noise process, as opposed to a deterministic transformation. The presence of the second argument, the prediction, makes our approach consistent with a large body of work in the sequence prediction literature, where the feedback is the reward. Decoupling the feedback and reward functions is the most notable feature of our approach.

Following a relatively recent trend in sequence prediction research (e.g. see [LW94, HKW95, Vov98, Sch99, CBFH+93, CBFHW93, HKW98, CBL99, FS97, ACBFS95, Vov99]) we make no assumptions whatsoever concerning the sequence to be predicted, meaning that we do not require, for instance, a statistical model of the sequence. For lack of a model, we need to assume that the sequence is arbitrary and therefore generated by an all-powerful device or adversary, which, among other things, is aware of the strategy a prediction algorithm is using. It might seem that competing with such a powerful opponent is hopeless. This is why, instead of the absolute performance of a prediction algorithm, it is customary to consider the regret w.r.t. the best predictor in some class. In this paper we make the choice of comparing our algorithm against the best constant predictor. Even if it seems a very restrictive setting, let us remind the reader that the best constant prediction is picked after the whole sequence is known, that is with a much better knowledge than any prediction algorithm has available and even more so in the incomplete feedback setting. Moreover, constant predictors are the focus of an important line of research on iterated games [Han57, FS97, ACBFS95]. Finally, the result can be readily extended to a finite class of arbitrary predictors along the same lines of [ACBFS95]. The details of this extension will be included in future versions of this paper.

Our research is closely related to the the one presented in [FS97], where the subject is, indeed, the problem of learning a repeated game from the point of view of one of the players —which can be thought of, indeed, as a predictor, once we accept that prediction can be rewarded in general ways and not according to a metric. In that work the authors designed the Multiplicative Weighting algorithm and proved that it has regret $O(\sqrt{T})$ when compared against the optimal constant strategy. This algorithm is used as a subroutine of ours. In their setting the predictor receives as input not the sequence at past rounds but the rewards every alternate prediction (not only the one made) would have received. Since this is all that matters to their algorithm, this setting is called *full information game* in [ACBFS95], even if, according to our definitions, the sequence and not the reward is the primary information. In the latter paper, a *partial information game* corresponds to the multi-armed bandit problem, in which only the reward relative to the prediction made is known to the predictor. What would have happened picking any of the other choices remains totally unknown. The best bound on the regret for this problem has been recently improved to $O(\sqrt{T})$ [Aue00].

In the present work we extend this result to our more general setting, provided that the feedback and loss functions jointly satisfy a simple but non-trivial condition. This case includes relevant special cases, such as the bandwidth allocation and quality control problems mentioned at the beginning of the present section, as well the classic multi-armed bandit problem and others. In this case it is possible to prove a bound of

$O(T^{3/4})$ on the regret. The aforementioned condition is not specific to our algorithm: indeed we proved that, when it is not satisfied, any algorithm would incur a regret $\Omega(T)$, just as a prediction with no feedback at all.

Also closely related is the work presented in [WM00], where the same worst case approach to sequence prediction is assumed, but the sequence is available to a prediction algorithm only through noisy observations. Albeit very general, their results make some assumptions on the noise process, such as statistical independence between the noise components affecting observations at different time steps. Our feedback model encompasses also the situation of noisy observations, but gives up any statistical assumptions on the noise process too, in analogy with the notion of "malicious errors" in the context of PAC learning ([KL93]). That is we claim our work can be seen also as a worst case approach to the prediction of a noisy sequences.

The report is structured as follows. In Section 2 we formally describe the problem. In Section 3 we describe the basic algorithm and prove bounds on its performance. In Section 4 we review some examples and highlight some shortcomings of the basic algorithm and show how to overcome them. In Section 5 we present a general algorithm and prove that the algorithm is essentially the most general. In Section 6 we discuss our results.

# 2 The Model

We describe the problem as a game between a player choosing an action $g_t$ and an adversary choosing the action $y_t$ at time $t$. There are $K$ possible actions available to the player, w.l.o.g., from the set $[K] = \{1, \dots, K\}$, and $R$ actions in the set $[R]$ from which the adversary can pick from. At every time step the player suffers a loss equal to $\ell(y_t, g_t) \in [0, 1]$.

The game is played in a sequence of trials $t = 1, 2, \dots, T$. The adversary has full information about the history of the game, whereas the player only gets a feedback according to the function $f(y, g)$. Hence the $R \times K$-matrices $L$ and $F$, with $L_{ij} = \ell(i, j)$ and $F_{ij} = f(i, j)$ completely describe an instance of the problem. At each round $t$ the following events take place.

1. The adversary selects an integer $y_t \in [R]$.

2. Without knowledge of the adversary's choice, the player chooses an action by picking $g_t \in [K]$ and suffers a loss $x_{g_t}(t) = \ell(y_t, g_t)$.

3. The player observes $f_t = f(y_t, g_t)$.

Note that due to the introduction of the feedback function this is a generalization of the partial information game of [ACBFS95].

Let $W(T) := \sum_{t=1}^{T} x_{g_t}(t) = \sum_{t=1}^{T} \ell(y_t, g_t)$ be the total loss of player $A$ choosing $g_1, \ldots, g_T$. We measure the performance of the player by the expected *regret* $R_A$, which is the difference between the total loss of $A$ and the total loss of the best constant choice $g^*$, that is $\sum_{t=1}^{T} x_{g_*}(t)$.

$$R_A := \max_{y_1,\ldots,y_T} \mathbf{E}\left[\sum_{t=1}^{T} \ell(y_t, g_t) - \min_j \sum_{t=1}^{T} \ell(y_t, j)\right]$$

where each $y_i$ is a function of $g_1, \ldots, g_{t-1}$. In some works the corresponding min-max problem is investigated, transforming the loss into a reward. The two settings are equivalent, as it is easy to check.

# 3 The Basic Algorithm

For the full information case the following *Multiplicative Weighting Algorithm* (see Fig. 1) has been used in different settings and has been recently analyzed in [ACBFS95]. Fig. 2 shows the *Hedge Algorithm* which is their setting. The following Lemma shows the equivalence of both algorithms.

**Lemma 1** *Hedge and MW are equivalent algorithms.*

**Proof:** By induction, we prove that in each round the probability $p_i(t)$ computed by MW is identical to the probability $\tilde{p}_i(t)$ of the Hedge algorithm.

Note that for all $i, j \in [K]$ we have for the MW algorithm

$$\frac{p_i(t)}{p_j(t)} = \frac{p_i(t-1)}{p_j(t-1)} \frac{\exp(\eta x_j(t))}{\exp(\eta x_i(t)} = \frac{\exp(\eta L_j(t))}{\exp(\eta L_i(t))} .$$

We have for the hedge algorithm

$$\frac{\tilde{p}_i(t)}{\tilde{p}_j(t)} = \frac{\exp(\eta L_j(t))}{\exp(\eta L_i(t)} .$$

**Multiplicative Weighting Algorithm (MW)**
**input** $K$
**constant** $\eta \in (0,1)$
**begin**
  Initialize $p_i(1) := \frac{1}{K}$ for all $i \in \{1,\ldots,K\}$.
  **for** $t$ **from** $1$ **to** $T$ **do**
    Choose $g_t$ according to probabilities $p(t)$.
    Receive the loss vector $x(t)$
    $Z_t := \sum_{i=1}^{K} \frac{p_i(t)}{\exp(\eta x_i(t))}$
    **for** $i$ **from** $1$ **to** $K$ **do**
      $$p_i(t+1) := \frac{p_i(t)}{\exp(\eta x_i(t)) Z_t}$$
    **od**
  **od**
**end**

Figure 1: The multiplicative weighting algorithm.

**Hedge Algorithm**
**input** $K$
**constant** $\eta \in (0,1)$
**begin**
  $p_i(1) := \frac{1}{K}$, for all $i \in [K]$.
  $L_i(0) := 0$, for all $i \in [K]$.
  **for** $t$ **from** $1$ **to** $T$ **do**
    Choose $g_t$ according to probabilities $p_i(t)$.
    Receive the loss vector $x(t)$
    **for** $i$ **from** $1$ **to** $K$ **do**
      $L_i(t) := L_i(t-1) + x_i(t)$
    **od**
    $W_t := \sum_{i=1}^{K} \frac{1}{\exp(\eta L_i(t-1))}$
    **for** $i$ **from** $1$ **to** $K$ **do**
      $$p_i(t+1) := \frac{1}{\exp(\eta L_i(t)) W_t}$$
    **od**
  **od**
**end**

Figure 2: The hedge algorithm.

■

This Lemma implies $\prod_{i=1}^{t} Z_i = W_t$, since MW defines $p_i(t+1) = \frac{\exp(-\eta L_i(t))}{\prod_{i=1}^{t} Z_i}$ and hedge defines $p_i(t+1) = \frac{\exp(-\eta L_i(t))}{W_t}$.

The analysis of [FS97] leads to a tight result for the full knowledge model. We will base our analysis on an adaption of their main theorem. Let us define

$$\Phi(x) \ := \ e^x - 1 - x.$$

**Lemma 2** *For all* $z \in [-1, 1]$*:*

$$\frac{z^2}{e} \ \leq \ \Phi(z) \ \leq \ (e-2)z^2$$

**Proof:** It is straightforward. ■

**Lemma 3** *For all* $\eta \in [-1, 1]$*, and for all* $x \in [-1, 1]$*:*

$$1 + \eta x + \frac{1}{2}\Phi(\eta)x^2 \quad \leq \quad e^{\eta x} \quad \leq \quad 1 + \eta x + 2\Phi(\eta)x^2 \ .$$

**Proof:** The proof uses Lemma 2 for $z \in [-1, 1]$:

$$\frac{z^2}{e} \ \leq \ e^z - 1 - z \ \leq \ (e-2)z^2 \ .$$

Note that $\eta x \in [-1, 1]$. Then, we have

$$\frac{1}{2}\Phi(\eta)x^2 \ \leq \ \frac{1}{e(e-2)}x^2(e^\eta - 1 - \eta) \ \leq \ \frac{1}{e}x^2\eta^2 \ \leq \ e^{\eta x} - 1 - \eta x$$

and

$$e^{\eta x} - 1 - \eta x \ \leq \ (e-2)x^2\eta^2 \ \leq \ e(e-2)x^2(e^\eta - 1 - \eta) \ \leq \ 2\Phi(\eta)x^2 \ .$$

■

The following theorem establishes a bound on the performance of MW that holds for any loss function $\ell$.

**Theorem 1** *For* $\eta \in (0, 1)$*, for any loss matrix* $L$ *with* $R$ *rows and* $K$ *columns with entries in the range* $[0, 1]$ *and for any sequence* $y_1, \ldots, y_T$ *the sequence of* $p(1), \ldots, p(T)$ *produced by algorithm MW satisfies*

$$\sum_{t=1}^{T}\sum_{i=1}^{K} \ell(y_t, i)p_i(t) \ \leq \ \min_j \sum_{t=1}^{T} \ell(y_t, j) + \frac{2\Phi(\eta)}{\eta}\sum_{t=1}^{T}\sum_{i=1}^{K} p_i(t)\ell(y_t, i)^2 + \frac{\ln K}{\eta} \ .$$

**Proof:** We will prove that, for any choice of $q(1), \dots, q(T)$ where $q(t)$ represents a probability distribution over $[R]$ and $q_i(t)$ is the probability of action $i$ in step $t$ and for any vector $p$ representing a probability distribution over $[K]$, we have:

$$\sum_{t=1}^{T} q_t^T \; L\, p(t) \leq \sum_{t=1}^{T} q_t^T \; L\, p + \frac{2\Phi(\eta)}{\eta} \sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) (q_t^T \; L\, e_i)^2 + \frac{\ln K}{\eta}$$

Choosing $q_t$ and $p$ to be unit vectors leads to the claim.

We use the *Kullback-Leibler divergence*, also called *relative entropy*, which is defined for probability distributions over $\{1, \dots, K\}$ by

$$\mathrm{RE}(P \,||\, Q) \;:=\; \sum_{i=1}^{K} P_i \ln \left( \frac{P_i}{Q_i} \right) \; .$$

**Lemma 4** *For any iteration $t$ where MW is used with parameter $\eta > 0$ and for any probability vector $p$*

$$\begin{aligned}\mathrm{RE}(p \,||\, p(t+1)) - \mathrm{RE}(p \,||\, p(t)) \quad &\geq \quad \eta\, q^T(t) L p - \eta\, q^T(t) L p(t) \\ &\quad +2\Phi(\eta) \sum_{i=1}^{K} p_i(t) \; \left( q(t)^T L e_i \right)^2\end{aligned}$$

**Proof:**

$$\begin{aligned}
&\mathrm{RE}(p \,||\, p_{t+1}) - \mathrm{RE}(p \,||\, p(t)) \\
&\quad = \sum_{i=1}^{K} p_i \ln \frac{p_i}{p_i(t+1)} - \sum_{i=1}^{K} p_i \ln \frac{p_i}{p_i(t)} \\
&\quad = \sum_{i=1}^{K} p_i \ln \frac{p_i(t)}{p_i(t+1)} \\
&\quad = \sum_{i=1}^{K} p_i \ln Z_t \exp(\eta\, q(t)^T L e_i) \\
&\quad = \eta \left( \sum_{i=1}^{K} p_i \; q(t)^T L e_i \right) + \ln Z_t \\
&\quad = \eta\, q(t)^T L p + \ln \left( \sum_{i=1}^{K} \frac{p_i(t)}{\exp(\eta x_i(t))} \right) \\
&\quad \leq \eta\, q(t)^T L p + \ln \left( \sum_{i=1}^{K} p_i(t) \left( 1 - \eta x_i(t) + 2\Phi(\eta) x_i(t)^2 \right) \right)
\end{aligned}$$

$$\leq \quad \eta\, q(t)^T Lp + \ln\left(1 - \sum_{i=1}^{K} \eta p_i(t) x_i(t) + \sum_{i=1}^{K} 2 p_i(t)\Phi(\eta) x_i(t)^2\right)$$

$$\leq \quad \eta\, q(t)^T Lp - \eta q(t)^T Lp(t) + \sum_{i=1}^{K} 2 p_i(t)\ \Phi(\eta)\ \left(q(t)^T L e_i\right)^2$$

The last lines use the fact that $\ln(1+x) \leq x$ for any $x > -1$ and Lemma 3. ■

We sum over $t$:

$$\eta \sum_{t=1}^{T} q(t)^T Lp(t) - \eta \sum_{t=1}^{T} q(t)^T Lp - 2\Phi(\eta) \sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) \left(q(t)^T L e_i\right)^2$$
$$\leq \quad \mathrm{RE}(p \,||\, p(1)) - \mathrm{RE}(p \,||\, p(T+1))$$

Noting that $\eta > 0$, $\mathrm{RE}(p \,||\, p(t+1) \geq 0$, and $0 \leq \mathrm{RE}(p \,||\, p(1)) \leq \ln K$ for all $p$ gives the claim. ■

Our algorithm relies on the existence of a $K \times K$ matrix $G$ satisfying the following equation:

$$F\, G = L\ .$$

For instance, if $F$ is non singular, this property can be fulfilled by choosing $G = F^{-1}L$. If such a $G$ does not exist the basic algorithm fails, i.e. it cannot compute a strategy at all.

The algorithm can be described as follows. First, it estimates the loss vector using the matrix G and the feedback. This estimate is fed into the MW algorithm which returns a probability distribution on the player's actions. MW tends to choose an action with very low probability if the associated loss over the past history of the game is high. This is not acceptable in the partial information case, because actions are useful also from the point of view of the feedback. Therefore, and again in analogy with [FS97], the algorithm adjusts the distribution $p(t)$, output by the MW algorithm, to a new distribution $\hat{p}(t)$ such that $\hat{p}_i(t) \geq \frac{\gamma}{K}$ for each $i$. We will give an appropriate choice of $\gamma$ and other parameters affecting the algorithm later on. What is new to this algorithm and what makes it much more general, is the way the quantities $x_i(t)$ are estimated. More in detail, given $F$ and $L$ and assuming there is a $G$ such that $FG = L$, our basic algorithm works as shown in Fig. 3.

The following lemma shows that $\hat{x}(t)$ is an unbiased estimator of the loss vector $x(t)$.

**FeedExp3**
**input** $F, L$
**begin**
  Compute $G$ such that $F\,G = L$.
  Choose $\eta, \gamma \in (0,1)$ according to $G$.
  Initialize $p(1)$ with $p_i(1) := \frac{1}{K}$ for all $i \in \{1, \ldots, K\}$.
  **for** $t$ **from** $1$ **to** $T$ **do**
    Select action $g_t$ to be $j$ with probability
      $\hat{p}_j(t) := (1-\gamma)p_j(t) + \frac{\gamma}{K}$.
    Receive as feedback the number $f_t$.
    **for** $i$ **from** $1$ **to** $K$ **do**
      $$\hat{x}_i(t) := \frac{f_t\, G_{g_t,i}}{\hat{p}_{g_t}(t)}$$
    **od**
    $Z_t := \sum_{i=1}^{K} p_i(t) \exp(\eta \hat{x}_i(t))$
    **for** $i$ **from** $1$ **to** $K$ **do**
      $$p_i(t+1) := p_i(t)\, \frac{\exp(\eta \hat{x}_i(t))}{Z_t}$$
    **od**
  **od**
**end**

Figure 3: The feedback exponential exploration and exploitation algorithm.

**Lemma 5** *For all $i, t$ we have*

$$\mathbf{E}[\hat{x}_i(t)|g_1, \ldots, g_{t-1}] = x_i(t) \quad \textit{and} \quad \mathbf{E}[\hat{x}_i(t)] = \mathbf{E}[x_i(t)] \; .$$

**Proof:** Note that

$$\mathbf{E}[\hat{x}_i(t)|g_1, \ldots, g_{t-1}] = \sum_{j=1}^{K} \hat{p}_j(x) \frac{F_{y_t,j}}{\hat{p}_j(x)} G_{j,i} = \sum_{j=1}^{K} F_{y_t,j} G_{j,i} = L_{y_t,i} = x_i(t) \; .$$

This implies

$$\mathbf{E}[\hat{x}_i(t)] = \mathbf{E}[\mathbf{E}[\hat{x}_i(t)|g_1, \ldots, g_{t-1}]] = \mathbf{E}[x_i(t)] \; .$$

■

Let $S_{y,i}(g) := F_{y,g} G_{g,i}$, for all $y, i, g \in \{1, \ldots, K\}$, $S^+ := \max_{y,g,i}\{S_{y,i}(g)\}$, $S^- := \min_{y,g,i}\{S_{y,i}(g)\}$, $\sigma := \max 0, -S^-$ and $\rho := S^+ - S^-$.

**Lemma 6** *For any sequence $y_1, \ldots, y_T$ the sequence $\hat{p}(1), \ldots, \hat{p}(T)$ produced by FeedExp3 satisfies for all $j$:*

$$\sum_{t=1}^{T} \sum_{i=1}^{K} \hat{x}_i(t) \hat{p}_i(t) \;\le\; \sum_{t=1}^{T} \hat{x}_j(t) + \frac{2\Phi(\eta)\rho K T}{\eta\gamma} + \frac{\rho K \ln K}{\gamma\eta} + \frac{\gamma}{K} \sum_{t=1}^{T} \sum_{i=1}^{K} \hat{x}_i(t) \; .$$

**Proof:** Consider a game where $p(t)$ denotes the probability distribution and the estimated loss $\hat{x}(t)$ is the real loss. Then, the FeedExp3-algorithm above reduces to a MW-algorithm, where $x(t)$ is replaced by $\hat{x}(t)$. Note that the range of the estimation vector now is $[KS^-/\gamma, KS^+/\gamma]$. So, we normalize the loss by defining $\hat{x}'_i(t) := \frac{\gamma}{K\rho}\hat{x}_i(t) - S^-/\rho$ to $[0, 1]$. Theorem 1 now implies

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) \hat{x}'_i(t) \quad &\le \quad \sum_{t=1}^{T} \hat{x}'_j(t) + \frac{2\Phi(\eta)}{\eta} \sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) \hat{x}'_i(t)^2 + \frac{\ln K}{\eta} \\
&\le \quad \sum_{t=1}^{T} \hat{x}'_j(t) + \frac{2T\Phi(\eta)}{\eta} + \frac{\ln K}{\eta}
\end{aligned}$$

Rescaling leads to this inequality:

$$\sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) \hat{x}_i(t) \;\le\; \sum_{t=1}^{T} \hat{x}_j(t) + \frac{2\Phi(\eta)\rho K T}{\gamma\eta} + \frac{\rho K \ln K}{\gamma\eta}$$

In algorithm FeedExp3 we have defined $\hat{p}_i(t) := (1-\gamma)p_i(t) + \frac{\gamma}{K}$. Hence, we can apply

$$\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{p}_i(t)\hat{x}_i(t) = (1-\gamma)\left(\sum_{t=1}^{T}\sum_{i=1}^{K}p_i(t)\hat{x}_i(t)\right) + \frac{\gamma}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{x}_i(t) .$$

So, a choice $\gamma > 0$ implies the claim. ∎

**Lemma 7** *Let $\lambda > 0$ and $\delta > 0$. Then with probability at least $1-\delta$, for every action $i$, we have*

$$\sum_{t=1}^{T}\hat{x}_i(t) \geq \left(1-\frac{\Phi(\lambda)}{\lambda}\right)\sum_{t=1}^{T}x_i(t) - \frac{\rho K\ln(K/\delta)}{\gamma\lambda} - \frac{\Phi(\lambda)\sigma TK}{\gamma\lambda\rho} \qquad (1)$$

$$\sum_{t=1}^{T}\hat{x}_i(t) \leq \left(1+\frac{\Phi(\lambda)}{\lambda}\right)\sum_{t=1}^{T}x_i(t) + \frac{\rho K\ln(K/\delta)}{\gamma\lambda} + \frac{\Phi(\lambda)\sigma TK}{\gamma\lambda\rho} . \qquad (2)$$

**Proof:** We use a martingale argument in close analogy to the proof of Lemma 5.1 in [ACBFS95].

(1): Let us define the random variable

$$Z_t = \exp\left(\lambda\sum_{t'=1}^{t}(x'_i(t') - \hat{x}'_i(t')) - \Phi(\lambda)\sum_{t'=1}^{t}x'_i(t')\right)$$

where $x'_i(t) := \frac{\gamma}{K\rho}x_i(t) - S^-/\rho$ and $x'_i(t) := \frac{\gamma}{K\rho}\hat{x}_i(t) - S^-/\rho$ are normalized replacements of $x_i(t)$ and $\hat{x}_i(t)$. The main claim of the proof is that $\mathbf{E}[Z_T] \leq 1$. Given this claim, we have by Markov's inequality that

$$P[Z_T > K/\delta] \leq \delta/K$$

which, by some algebra, is equivalent to (1). We prove that $\mathbf{E}[Z_t] \leq 1$ for $t = 0,\ldots,T$ by induction on $t$ using a method given by Neveu ([Nev75], Lemma VII-2-9). For $t=0$, $Z_0 = 1$. To prove the inductive step for $t > 0$, we have that

$$\begin{aligned}\mathbf{E}[Z_t \mid g_1,\ldots,g_{t-1}] \quad &= \quad Z_{t-1}\exp\left(-\Phi(\lambda)x'_i(t)\right)\\ &\quad\quad \mathbf{E}[\exp(\lambda(x'_i(t) - \hat{x}'_i(t))) \mid g_1,\ldots,g_{t-1}] .\end{aligned}$$

Due to normalization we have $x'_i(t) \in [0,1]$ and $\hat{x}'_i(t) \in [0,1]$. Therefore $x'_i(t) - \hat{x}'_i(t) \leq 1$ and it follows by Lemma 3

$$\mathbf{E}[\exp(\lambda(x'_i(t) - \hat{x}'_i(t))) \mid g_1,\ldots,g_{t-1}]$$

$$
\begin{aligned}
&\leq \mathbf{E}[1 + \lambda(x'_i(t) - \hat{x}'_i(t)) + \Phi(\lambda)(x'_i(t) - \hat{x}'_i(t))^2 |\, g_1, \ldots, g_{t-1}] \\
&\leq 1 + \Phi(\lambda) x'_i(t) \\
&\leq \exp(\Phi(\lambda) x'_i(t)) \ .
\end{aligned}
$$

The second inequality follows from Lemma 5 and the following chain of inequalities:

$$
\begin{aligned}
\mathbf{E}[(x'_i(t) - \hat{x}'_i(t))^2 \mid g_1, \ldots, g_{t-1}] &= \mathbf{E}[\hat{x}'_i(t)^2 \mid g_1, \ldots, g_{t-1}] - x'_i(t)^2 \\
&\leq \mathbf{E}[\hat{x}'_i(t)^2 \mid g_1, \ldots, g_{t-1}] \\
&\leq \mathbf{E}[\hat{x}'_i(t) \mid g_1, \ldots, g_{t-1}] \\
&= x'_i(t) \ .
\end{aligned}
$$

(2): We define the random variable

$$
Z_t = \exp\left( \lambda \sum_{t'=1}^{t} (\hat{x}'_i(t') - x'_i(t')) - \ \Phi(\lambda) \sum_{t'=1}^{t} x'_i(t') \right) \ .
$$

Again the main claim of the proof is that $\mathbf{E}[Z_T] \leq 1$ using claim, we have by Markov's inequality with $P[Z_T > K/\delta] \leq \delta/K$, which, by some algebra, is equivalent to (2). The rest of this proof is analogous to the first part.

■

We single out a special case of Lemma 7 for further reference.

**Corollary 1** *For $\gamma, \delta > 0$ and for the random variable $g^* = \arg\max_j \sum_{t=1}^{T} x_j(t)$ we have with probability $1 - \delta$:*

$$
\sum_{t=1}^{T} \hat{x}_{g^*}(t) \quad \leq \quad \left(1 + \frac{\Phi(\lambda)}{\lambda}\right) \sum_{t=1}^{T} x_{g^*}(t) + \frac{\rho K \ln(K/\delta)}{\gamma\lambda} + \frac{\Phi(\lambda)\sigma T K}{\gamma\lambda\rho} \ .
$$

**Theorem 2** *If there exists $G$ such that $F\,G = L$ then the expected regret $\mathbf{E}[R_{\mathit{FeedExp3}}(T)]$ of algorithm FeedExp3 after $T$ steps is bounded by*

$$
\mathbf{E}[R_{\mathit{FeedExp3}}(T)] \ = \ O(T^{3/4} (\ln T)^{1/2} K^{1/2})
$$

*with a constant factor linear in $\rho + \frac{\sigma}{\rho}$.*

**Proof:** We first rewrite the expected loss $\mathbf{E}[W(T)]$ of algorithm FeedExp3 in a different way:

$$\begin{aligned}
\mathbf{E}[W(T)] &= \sum_{t=1}^{T} \mathbf{E}[x_{g_t}] \\
&= \sum_{t=1}^{T} \mathbf{E}[\mathbf{E}[x_{g_t}|g_1 \ldots g_{t-1}]] \\
&= \sum_{t=1}^{T} \mathbf{E}\left[\sum_{i=1}^{K} x_i(t)\hat{p}_i(t)\right] \\
&= \mathbf{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{K} x_i(t)\hat{p}_i(t)\right] \qquad (3)
\end{aligned}$$

We then apply Lemma 7(2) and choose $\gamma = \frac{K^{1/2}(\ln T)^{1/2}}{T^{1/4}}$, $\lambda = \frac{1}{T^{1/2}}$ and $\delta = \frac{1}{T^2 K}$. Lemma 2 states for $z \in [-1,1]$ that $\frac{z^2}{e} \leq \Phi(z) \leq (e-2)z^2$ and Lemma 3 for $z \in (0,1] \,:\, (1-z)^{-1} \leq 1+2z$. This implies for $\gamma \in ]0,1[$:

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{i=1}^{K} \hat{p}_i(t)x_i(t) &\leq \frac{\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{p}_i(t)\hat{x}_i(t) + \frac{\rho K \ln(K/\delta)}{\gamma\lambda} + \frac{\Phi(\lambda)\sigma T K}{\gamma\lambda\rho}}{1-\frac{\Phi(\lambda)}{\lambda}} \\
&\leq \left(1 + 2\frac{\Phi(\lambda)}{\lambda}\right) \\
&\quad \left(\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{p}_i(t)\hat{x}_i(t) + \frac{\rho K \ln(K/\delta)}{\gamma\lambda} + \frac{\Phi(\lambda)\sigma T K}{\gamma\lambda\rho}\right) \\
&\leq \left(1 + \frac{2}{T^{1/2}}\right)\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{p}_i(t)\hat{x}_i(t) + \\
&\quad \left(\rho + \frac{\sigma}{\rho}\right) T^{3/4}(\ln T)^{1/2}K^{1/2} + o(T^{1/2}K^{1/2}) \qquad (4)
\end{aligned}$$

Choose $\eta = T^{-1/2}$. Then Lemma 6 implies, for all $j$:

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{p}_i(t)\hat{x}_i(t) &\leq \sum_{t=1}^{T}\hat{x}_j(t) + \frac{2\Phi(\eta)\rho K T}{\eta\gamma} + \frac{\rho K \ln K}{\gamma\eta} + \frac{\gamma}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{x}_i(t) \\
&\leq \sum_{t=1}^{T}\hat{x}_j(t) + \frac{3\rho T^{3/4}K^{1/2}}{(\ln T)^{1/2}} + \frac{\gamma}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{x}_i(t)\,. \qquad (5)
\end{aligned}$$

Note that from Lemma 7(1) it follows for the rightmost additional term with probability $1-\delta := 1-\frac{1}{T^2K}$.

$$\begin{aligned}\frac{\gamma}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{x}_i(t) &\leq \frac{\gamma}{K}\left(1+\frac{\Phi(\lambda)}{\lambda}\right)\sum_{t=1}^{T}\sum_{i=1}^{K}x_i(t)+\frac{\rho K\ln(K/\delta)}{\lambda}+\frac{\Phi(\lambda)\sigma TK}{\lambda\rho}\\ &\leq O(T^{3/4}(\ln T)^{1/2}K^{1/2})+O((\rho+\tfrac{\sigma}{\rho})T^{1/2}(\ln T)K)\ . \qquad (6)\end{aligned}$$

At last we will use Corollary 1. Then we have with probability $1-\delta$.

$$\begin{aligned}\sum_{t=1}^{T}\hat{x}_{g^*}(t) &\leq \left(1+\frac{\Phi(\lambda)}{\lambda}\right)\sum_{t=1}^{T}x_{g^*}(T)+\frac{\rho K\ln(K/\delta')}{\gamma\lambda}+\frac{S^-TK\Phi(\lambda)}{\rho\gamma\lambda}\\ &\leq \left(1+\frac{1}{T^{-1/2}}\right)\sum_{t=1}^{T}x_{g^*}(T)\\ &\quad +O((\rho+\tfrac{-S^-}{\rho})T^{3/4}(\ln T)^{1/2}K^{1/2}) \qquad (7)\end{aligned}$$

For the expectation we add error term $3\delta T$ for the combined error probabilities $3\delta$ and combine (3), (4), (5), (6), and (7).

$$\begin{aligned}\mathbf{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{p}_i(t)x_i(t)\right] &\leq \mathbf{E}\left[\sum_{t=1}^{T}x_{g^*}(t)\right]\left(1+\frac{4}{T^{1/2}}\right)+3\delta T\\ &\quad +O((\rho+\tfrac{-S^-}{\rho})T^{3/4}(\ln T)^{1/2}K^{1/2})\\ &\leq \mathbf{E}\left[\sum_{t=1}^{T}x_{g^*}(t)\right]+\ O((\rho+\tfrac{-S^-}{\rho})T^{3/4}(\ln T)^{1/2}K^{1/2})\end{aligned}$$

■

# 4 Applications, Limits and Extensions

We are now equipped to show how the bandwidth allocation problem that initially prompted this research, as well as other important examples, can be solved using this algorithm, but we will also see that only some tweaking allows to solve even more prediction problems. We will see in the next section that these "tricks" lead to a general algorithm, that, after some preprocessing, uses the basic algorithm to achieve sub-linear regret whenever this is feasible.

In the bandwidth allocation problem the feedback function is defined as follows (*threshold feedback*):

$$f(y,g)=\begin{cases}0 \text{ if } y<g\\ 1 \text{ otherwise}\ .\end{cases}$$

The feedback matrix $F$ is therefore a lower triangular matrix with only 1's on the diagonal and below. Therefore, the condition $FG = L$ is satisfied for $G = F^{-1}L$.

The multi-armed bandit problem with partial information of [ACBFS95] corresponds to the case $F = L$. Under this condition, $G = I$ is a suitable choice. A somehow dual situation arises when $F = I$, that is when the feedback is a binary "hit or miss" information. Then $G = L$ is a suitable choice for $G$.

A more troublesome situation is the full feedback case. Even if in this case the machinery presented in this paper is not necessary, since an expected regret of $O(T^{1/2} \log K)$ can be achieved by the MW algorithm [FS97], it is clear that a general algorithm for this class of problems must be able to solve this special case, too. A natural choice for $F$ is $F_{ij} = i$, which implies $f_t = y_t$. Unfortunately, such a matrix has rank 1 and therefore the condition $FG = L$ can be satisfied only when L has a very special, and rather trivial, form. But more than the specific values of the entries of $F$, what defines "full feedback" is the fact that no two entries in every column of $F$ have the same value, that is there is a bijection between the values in $F_i$ and the range of $y_t$. If $F$ satisfies this property, it is possible to compute $y_t$ from $f_t$ and hence we can say we are still in the full information case. Therefore, we are interested in finding a full rank matrix within the set of matrices just described, which all represent the full feedback case.

One possible solution is to replace every diagonal entry with a number large enough to satisfy Hadamard's theorem, that is:

$$|F_{ii}| > \sum_{j=1, j \neq i}^{K} |F_{ij}| \, .$$

implies that $\det(F) \neq 0$. But this solution is specific to the full feedback case, whereas the problem of singular or low rank $F$ arises in many contexts.

For instance, consider the threshold feedback and modify slightly the definition to be $f(y, g) = 0$, if $y \leq g$ and 1 otherwise. Then $F$ becomes singular, but it is enough to reverse the arbitrary roles of 0 and 1 to get an equivalent problem, where this time $F$ is invertible.

An acceptable transformation of $F$ can be detailed as a set of functions for every column of $F$, from the range of the elements of $F$ into some other range. The goal is to obtain a new matrix $F'$, where every column is obtained applying one of the functions to the elements of a column of $F$, for which there is a $G$ such that $F'G =$

$L'$. It is clear that $F'$ can have more columns than $F$, because every column can be transformed in different ways, but no fewer, since every action has to be represented. This corresponds to introducing new actions that are essentially replicas, but for each of which the feedback undergoes a different transformation. From the point of view of the loss, these additional actions are totally equivalent and therefore we need to extend $L$ into a larger matrix $L'$ by duplicating the appropriate columns. What we seek is a general way to expand $F'$ so as to keep the number of columns reasonably small but making the linear span of $F'$ all-inclusive, that is such that it cannot be enlarged by adding more columns obtained in a feasible way. This can be accomplished as follows. For every column $F_i$ containing $r_i$ distinct values (w.l.o.g. from the set $[r_i]$) we define $r_i$ columns $F'_{R_i+1} \dots F'_{R_i+r_i}$, where $R_i = \sum_{j=1}^{i-1} r_i$, as follows: $F'_{R_i+j,k} = [j = F_{i,k}]$, for $1 \leq j \leq r_i$, where $[P] = 1$ if $P$ is true and 0 otherwise. As to $L'$, we set $L'_j = L_i$ if and only if $R_i < j \leq R_i + r_i$. It is straightforward to check that the matrix $F'$ obtained this way has the largest possible linear span among all the ones that can be obtained from $F$ via the transformations detailed above. Also, since $F$ is $R \times K$, $F'$ is at most $R \times KR$. These are more columns than we need and would impact negatively the bounds on the regret: therefore we will pick the smallest subset of columns $S$ which is still good for our purposes, that is, that satisfies the following conditions:

- all the columns of $L$ are represented in $L'$ or, equivalently, all the actions in the original instance are represented, that is for every $i \in [K]$ there is a $j \in S$ such that $R_i < j \leq R_i + r_i$;
- $\mathcal{L}(\{F'_i : i \in S\}) = \mathrm{range}(F')$.

The final feedback and distance matrices can be obtained by dropping all the columns not in $S$ from $F'$ and $L'$, and we will continue to use the same symbols for the submatrices defined this way. In the next section we will present a greedy algorithm which solves this problem.

Let us see how this helps in the full feedback case. Recall that a natural choice for $F$ is $F_{ij} = i$. Therefore, the corresponding $F'$ has maximum rank (some columns of $F'$ form an $R \times R$ identity matrix), $F'G = L$ can be solved for $G$ and the general algorithm can be applied successfully.

A further complication arises from *non-exploitable* actions. These are actions which for any adversarial strategy do not turn out to be optimal. The problem here is that the condition $FG = L$ might be impossible to be satisfied because of some columns related to non-exploitable actions. Consider, for instance

$$F = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} .$$

Here columns 1 of $L$ is not in the linear span of $F$, but it is easy to see that actions 3 and 4 can be always preferred to the first. Therefore it might seem reasonable to simply drop the first column as it is related to a non-exploitable action. It turns out, though, it is just action 1 that provides the necessary feedback to estimate the loss. It is clear that simply omitting non-exploitable actions is not a good strategy.

As with the feedback matrix $F$, the solution for these problems is to transform the loss matrix $L$ into a new $L'$ in a way that does not lower the regret.

If we add the same vector $x$ to every column of $L$, we are not changing the problem instance in any substantial way, since the regret, our performance measure, is invariant w.r.t. this transformation. Therefore we are interested in those transformations that help fulfilling the condition $FG = L$. This time, it makes sense to try to obtain a matrix $L'$ from $L$ of minimum rank. Rank minimization is a difficult problem in general, but this special case turns out to be rather trivial.

**Lemma 8** *Given three matrices $L$, $L'$ and $L''$ such that for every $i$ $L'_i = L_i - L_j$ and $L''_i = L_i - x$, we have that, for any vector $x$ and index $j$, $\mathcal{L}(L') \subseteq \mathcal{L}(L'')$.*

**Proof:** Since $L_i - L_j = L_i - x - (L_j - x)$, the lemma follows. ■

Therefore choosing $x$ equal to one of the columns of $L$ minimizes the linear span of $L'$. In the following we will assume $L_1 = (0, \ldots, 0)$ w.l.o.g.

As to non-exploitable actions, we first need to formally define them. Let us define a partition[1] of the set of mixed strategies (for the adversary) as follows. Every element of the partition is centered around a column of $L'$ and is defined as:

$$N(L_i) = \{v \in \mathcal{A} \mid \forall j : \; L_i \neq L_j \Rightarrow vL_i \leq vL_j\}$$

where the set $\mathcal{A} := \{v \in [0,1]^R | \; \sum_i v_i = 1\}$ denotes all possible mixed strategies of the adversary.

That is an element of this partition is the set of mixed adversarial strategies such that a certain prediction is preferred to any other. If $N(L_i)$ is empty, then $i$ is a non-exploitable action. The rationale behind this definition is that no sensible algorithm will

[1]Strictly speaking, it is not a partition, but the idea helps the intuition

ever try this action for exploitation purposes (that is often), since there are other actions which bear a smaller loss. The interior of $N(L_i)$ is defined as follows:

$$S(L_i) = \{v \in \mathcal{A} \mid \forall j : \; L_i \neq L_j \Rightarrow vL_i < vL_j\}$$

The following lemma shows that we can replace every mixed adversarial strategy on the surface of some element of the partition by another strategy not on the surface, with no penalty in performance.

**Lemma 9** *For all mixed adversarial strategies $q \in \mathcal{A}$ there exists a column $L_i$ with $S(L_i) \neq \emptyset$ such that $q \in N(L_i)$.*

**Proof:** We concentrate on elements in the set $\mathcal{F} := \bigcup_i N(L_i) \setminus S(L_i)$. Note that we have

$$\mathcal{F} \;\subseteq\; \bigcup_{i,j} \{v \in \mathcal{A} \mid v(L_i - L_j) = 0\} \, .$$

Therefore $\mathcal{F}$ is a subset of a union of at most $K^2$ subspaces of dimension $R-2$. Since $\mathcal{A}$ is a $R-1$ dimensional polytope, the $\epsilon$-ball surrounding of a point $v \in N(L_i)$ contains elements not in $\mathcal{F}$.

Such an element $v' \notin \mathcal{F}$ is contained in a set $L_{i'}$ with $S(L_{i'}) \neq \emptyset$. Observe that in the limit ($\epsilon \to 0$) the loss of $v'L_j$ equals $vL_j$ for any column $j$. Thus, we have $vL_i = vL_{i'}$ and $v \in L_{i'}$. ∎

Hence, we can extend the definition of non-exploitable action to columns with $S(L_i) = \emptyset$, since their choice gives no improvement over actions with $S(L_i) \neq \emptyset$.

In order to extend the applicability of the basic algorithm, we set all the entries in the columns corresponding to non-exploitable actions equal to the size of the maximum element in its column in $L$. This can only increase the regret w.r.t. the best constant strategy, because none of the actions associated to these columns can be part of any optimal strategy. Furthermore, it is easy to check that the columns obtained this way are in the linear span of $F'$ for every $F$.

# 5 The General Algorithm

In Fig. 4 we show how to implement the construction of $F'$ and $L'$. Let $[F_{i,j} = v]_{i=1,\dots,R}$ denote the vector obtained replacing, in the $j$th column of $F$, every entry equal to $v$ by 1 and all others by 0. The algorithm constructs $F'$ and $L'$ by appending columns derived from $F$ and $L$ to their right sides.

Augmented with this kind of preprocessing for the loss and feedback matrices, our algorithm covers all the examples we considered. A natural question is therefore whether the condition $F'G = L'$ is not only necessary for our algorithm to apply, but in general for any useful algorithm. The answer is positive, meaning that if the condition cannot be fulfilled, then any algorithm will undergo a loss $\Omega(T)$.

**Theorem 3** *For any prediction game $(F, L)$ we have either one of the following situations:*

- *The General Algorithm solves it with an expected regret of*

$$\mathbf{E}[R_{General}] \ \leq \ O(T^{3/4}(\ln T)^{1/2} \max(K, R^{1/2})) \ .$$

- *There is an adversarial strategy which causes any algorithm $A$ to produce a regret of $\Omega(T)$ with probability $1/2$.*

**Proof:** In the previous section, we have already seen that we can map a sequence of actions for the prediction game $(F', L')$ to the instance $F, L$ in a way that does not essentially increase the regret. This proves the first part of the theorem. We can rephrase the second part as follows:

> Given an instance of the prediction game $(F, L)$ let be $F'$ and $L'$ the matrices obtained through the transformations detailed in the previous section. If there is no $G$ such that $F'G = L'$, then any prediction algorithm will undergo a loss $\Omega(T)$.

We associate a graph $H = (V, E)$ to the partition $\{N(L'_1), \ldots, N(L'_k)\}$ by defining $V = \{L_i : S(L'_i) \neq \emptyset\}$ and $(L'_i, L'_j) \in E$ if and only if $L'_i = L'_j$ or the sets $N(L'_i)$ and $N(L'_j)$ share a facet, i.e. a face of dimension $R - 2$. Note that for all $i$ the set $N(L'_i)$ describes a polytope of dimension $R - 1$ or its interior $S(L'_i)$ is empty.

Let $\mathcal{L}(E)$ be the linear span of the set of differences between vectors at the endpoints of each edge in $E$. We have the following

**Lemma 10** $\mathcal{L}(E) = \mathcal{L}(\{L'_i : L'_i \in V\})$ .

```
The General Algorithm
  input R × K-matrices F, L
  begin
    z := 0
    for j from 1 to K do
      a := 0
      for all values v in F_i do
        if [F_{i,j} = v]_{i=1,...,R} ∉ 𝓛(F'_1, ..., F'_z) then
          a := 1
          z := z + 1
          F'_z := [F_{i,j} = v]_{i=1,...,R}
          L'_z := L_j
          h(z) := j
        fi
      od
    if a = 0 then
      z := z + 1
      F'_z := (0, ..., 0)
      L'_z := L_j
      h(z) := j
    fi
  od
  b := 0
  for i from 1 to z do
    if b = 0 and S(L'_i) ≠ ∅ then
      b := i
    fi
  od
  for i from 1 to z do
    L'_i := L'_i − L'_b
  od
  for i from 1 to z do
    if S(L'_i) = ∅ then
      L'_i := (max_{j'} {L'_{j'i}}, ..., max_{j'} {L'_{j'i}})
    fi
  od
  Perform FeedExp3(F', L') such that
    an action g_t is replaced by h(g_t)
end
```

Figure 4: The General Algorithm

**Proof:** For each $L'_i \in V$, $L'_i = L'_i - L'_{i_1} + L'_{i_1} - L'_{i_2} + \ldots + L'_{i_p} - L'_1$, where $(L'_i, L'_{i_1}, \ldots, L'_{i_p}, L'_1)$ is a path connecting $L'_i$ to $L'_1$, if such a path exists.

We need only to prove that $H$ is connected. Given the two vertices $L'_i$ and $L'_j$, we seek a path joining them. Consider the segment joining a point in the interior of $N(L'_i)$ to one in the interior of $N(L'_j)$. Since the set of mixed strategies is convex, every point in the segment is a mixed strategy. Let us pick an arbitrary orientation for this segment and consider the sequence of polytopes that share with the segment some interior point, and specifically two consecutive entries in the sequence, $N(L'_h)$ and $N(L'_k)$. If the segment goes from the first to the second through a facet, then the two corresponding vertices in the graph are joined by an edge. If not, that means that the two polytopes share only a face of dimension $R-3$ or lower, e.g. a vertex or an edge. In that case we need to pick a different point in, say, $N(L_j)$. This is always possible because $N(L_1)$ has dimension $R-1$ whereas the set of points collinear with the designated point in $N(L_i)$ and any point in any face of dimension $R-3$ or lower has dimension at most $R-2$. ■

Now, let us assume that there is no $G$ such that $F'G = L'$. This implies that there is $L'_i$ such that $L'_i \notin \mathcal{L}(F')$. Let us assume $S(L'_i) = \emptyset$. By definition of $L'$, $L'_i = \alpha(1, \ldots, 1)$ for some $\alpha$. This implies, by definition of $F'$, $L'_i \in \mathcal{L}(F')$, a contradiction. Therefore $S(L'_i) \neq \emptyset$ and, by lemma 10, $\mathcal{L}(E) \not\subseteq \mathcal{L}(F')$. Hence, for some $(L'_i, L'_j) \in E$, we have that $L'_i - L'_j \notin \mathcal{L}(F')$. Since the range of $F'$ is the orthogonal complement to the null space of $F'^T$ we have that, for some non-zero vector $n \in \mathrm{Ker}(F'^T)$, $\quad n(L'_i - L'_j) \neq 0$. Let $y$ be a point in the interior of the facet shared by $N(L'_i)$ and $N(L'_j)$. We have that $y + \alpha n$ and $y - \alpha n$ are both mixed strategies for some $\alpha$. They are indistinguishable from the point of view of any algorithm because $(y + \alpha n)F' = (y - \alpha n)F' = yF'$, but they correspond to different optimal actions, and the regret implied by making the wrong choice is $|\alpha n(L'_i - L'_j)|$. ■

# 6 Conclusion and Open Problems

We solve the problem of discrete loss and feedback online prediction games in its general setting, presenting an algorithm which, on average, has sub-linear regret against the best constant choice, whenever this is achievable.

In the full knowledge case, it is well known that the average per step regret is bounded by $O(T^{-1/2})$. In [ACBFS95] it is shown that, if the feedback is identical to the loss, there is a an algorithm whose average regret is bounded by $O(T^{-1/3})$ (omitting polylogarithmic terms), recently improved to $O(T^{-1/2})$ [Aue00]. In the present paper, we show that, for every "reasonable" feedback, the average per step regret is at most $O(T^{-1/4})$. Otherwise, no algorithm can do better than $\Omega(T)$.

While we proved that no algorithm can attain sub-linear regret on a larger class of problems than ours does, it is an open problem whether such general prediction games can be solved with a bound on the regret as good as the one obtained for the multi-armed bandit problem, in the most general setting or under some additional assumptions.

It is straightforward to transfer the upper bounds shown for the worst case regret against constant predictors to the finite pool of general predictors (a.k.a. "expert") model, in analogy with the argument of [ACBFS95], Section 7. However, the lower bound is not readily applicable to this case and therefore it is an open question whether our general algorithm achieves sub-linear regret whenever it is possible in this context.

Another interesting question is whether a uniform algorithm exists that works for any feedback and loss functions and achieves the best known performance for each feedback. Note that the algorithms presented in this work, even when given in input a feedback function corresponding to the "full knowledge" case, guarantees only an average per step regret of $O(T^{-1/4})$, whereas $O(T^{-1/2})$ is the best bound known.

## 7 Acknowledgements

We wish to thank Richard Karp for suggesting this line of research. Gadiel Seroussi, Marcelo Weinberg, Neri Merhav, Nicolò Cesa-Bianchi and Richard Karp provided invaluable feedback about the paper and pointers to the literature.

## References

[ACBFS95] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 322–331. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[Aue00] Peter Auer. Using upper confidence bounds for online learning. In *Proceedings of the 41th Annual Symposium on Foundations of Computer Science*, pages 270–279. IEEE Computer Society Press, Los Alamitos, CA, 2000.

[CBFH+93] Nicolò Cesa-Bianchi, Yoav Freund, David P. Helmbold, David Haussler, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice (extended abstract). In *Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 382–391, San Diego, California, May 1993.

[CBFHW93] Nicolò Cesa-Bianchi, Yoav Freund, David P. Helmbold, and Manfred Warmuth. On-line prediction and conversion strategies. In *EUROCOLT: EUROCOLT, European Conference on Computational Learning Theory, EuroCOLT,*. LNCS, 1993.

[CBL99] Nicolò Cesa-Bianchi and Gabor Lugosi. Minimax regret under log loss for general classes of experts. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*. ACM Press, 1999.

[FS97] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 1997. to appear.

[Han57] James Hannan. Approximation to bayes risk in repeated play. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume III, pages 97–139. Princeton University Press, 1957.

[HKW95] D. Haussler, J. Kivinen, and M. K. Warmuth. Tight worst-case loss bounds for predicting with expert advice. *Lecture Notes in Computer Science*, 904:69, 1995.

[HKW98] D. Haussler, J. Kivinen, and M. K. Warmuth. Sequential prediction of individual sequences under general loss functions. *IEEE Transactions on Information Theory*, 44, 1998.

[KL93] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM Journal on Computing,*, 22(4):807–837, August 1993.

[LW94] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1 February 1994.

[Nev75] J. Neveu. *Discrete-Parameter Martingale*. North Holland, 1975.

[PS00] A. Piccolboni and C. Schindelhauer. Discrete prediction games with arbitrary feedback and loss. Technical Report A-00-18, Schriftenreihe der Institute für Informatik und Mathematik, Universität Lübeck, October 2000.

[Sch99] Robert E. Schapire. Drifting games. In *Proc. 12th Annu. Conf. on Comput. Learning Theory*, pages 114–124. ACM Press, New York, NY, 1999.

[Vov98] V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, April 1998.

[Vov99] V. Vovk. Competitive on-line statistics. In *The 52nd Session of the International Statistical Institute*, 1999.

[WM00] T. Weissman and N. Merhav. Universal prediction of binary individual sequences in the presence of noise. accepted to IEEE Trans. Inform. Theory, September 2000.