On Delayed Prediction of Individual Sequences*

Marcelo J. Weinberger
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA

Erik Ordentlich†
GlobespanVirata, Inc.
Santa Clara, CA 95051, USA

Abstract

Prediction of individual sequences is investigated for cases in which the decision maker observes a delayed version of the sequence, or is forced to issue his/her predictions a number of steps in advance, with incomplete information. For finite action and observation spaces, it is shown that the prediction strategy that minimizes the worst-case regret with respect to the Bayes envelope is obtained through sub-sampling of the sequence of observations. The result extends to the case of logarithmic loss. For finite-state reference prediction strategies, the delayed finite-state predictability is defined and related to its non-delayed counterpart. As in the non-delayed case, an efficient on-line decision algorithm, based on the incremental parsing rule, is shown to perform in the long run essentially as well as the best finite-state strategy determined in hindsight, with full knowledge of the given sequence of observations. An application to adaptive prefetching in computer memory architectures is discussed.

Index Terms: Delayed prediction, sequential decision, on-line algorithms, general loss functions, Lempel-Ziv algorithm.

*Parts of this paper were presented at the 2000 Data Compression Conference, Snowbird, Utah, USA.
†Work partially done while this author was with Hewlett-Packard Laboratories, Palo Alto, California.

1 Introduction

The problem of predicting a binary sequence $x^n = x_1 x_2 \cdots x_n$, with the goal of achieving an expected number of prediction errors (or "loss") that approaches the loss of the best constant predictor, has received considerable attention over the last five decades. Here, the expectation is with respect to a possible randomization in the prediction strategy, and the loss of the best constant predictor is given by the *Bayes envelope*, $\min(n_0(x^n), n_1(x^n))$, where $n_a(x^n)$ denotes the number of occurrences in x^n of $a \in \{0, 1\}$. The problem was first studied in the framework of the *sequential decision problem* [1] and the *approachability-excludability theory* [2]. The *minimax* strategy, which minimizes the worst-case *regret* (i.e., the excess loss over the Bayes envelope) over all n-tuples, was devised by Cover [3]. Other predictors were proposed in [4], in a context where the "competing" reference strategy was finite-state (FS), rather than constant, and in [5] and [6], in the context of *prediction with expert advice*. The worst-case normalized regret of all these strategies vanishes at an $O(1/\sqrt{n})$ rate. In particular, Cover's minimax scheme yields the same regret over all sequences, its main asymptotic term being $\sqrt{n/(2\pi)}$.

The usual setting in prediction problems is that the on-line decision maker observes a prefix $x_1 x_2 \cdots x_t$ of x^n for each time instant t, $t = 0, 1, \cdots, n-1$ (we assume the horizon n is known), and makes a prediction $p_{t+1}(1|x^t) \in [0, 1]$. This prediction can be interpreted as the probability of choosing 1 in a randomized selection of the next bit x_{t+1}. Thus, the expected loss takes the form $1 - p_{t+1}(x_{t+1}|x^t)$. However, in many applications of practical interest, the on-line decision maker has access to a *delayed* version of the sequence, or is forced to make inferences on the observations a number of instants in advance. Such situations may arise when the application of the prediction is delayed relative to the observed sequence due to, e.g., computational constraints. The delay d, which is assumed known, affects the prediction strategy in that the prediction for x_{t+1} is now based on $x_1 x_2 \cdots x_{t-d}$ only. Since every such predictor is a particular case of a non-delayed one, the achievable performance (under any performance metric) cannot improve. On the other hand, the delay does not affect the performance of a constant predictor, so that the Bayes envelope is still our targeted loss. The question arises: How badly can the worst-case regret be affected by this delay?

At first glance it would appear that the effect of the delay is asymptotically negligible, mainly because the setting of competing against a constant strategy (for a given individual

sequence) is often associated to a probabilistic setting in which the data are drawn from a *memoryless* source. For a memoryless source, the expected loss incurred at time t for delayed prediction is the same as the expected loss that the predictor would incur, without delay, at time $t-d$. In addition, for an individual sequence, as t grows, the window of d "hidden" bits cannot significantly affect the statistics. Therefore, one would be inclined to ignore the delay and apply any of the above prediction schemes (namely, use at time t the same probability that the non-delayed predictor would have used at time $t-d$). As shown in Appendix A, application of the minimax strategy of [3] in such a manner indeed yields vanishing regret for all sequences, but it results in an asymptotic worst-case regret $2d+1$ times higher than in the non-delayed case. It is also shown in Appendix A that for a similar strategy based on the exponential weighting algorithm of [5] and [6], the worst-case normalized regret behaves asymptotically as $\sqrt{(2d+1)(\ln 2)/(2n)}$ (thus, the multiplicative factor over the $d=0$ case is $\sqrt{2d+1}$).[1]

The above additional regret due to the delay is immediately seen to be too high, once we realize that a simple "sub-sampling" strategy, used in conjunction with any of the above schemes for non-delayed prediction, yields a multiplicative factor of only $\sqrt{d+1}$ in the worst-case regret. Specifically, if we sub-sample the original sequence x^n at a rate $1/(d+1)$, and process the resulting $d+1$ sub-sequences separately, each sample x_{t+1} is predicted based only on previous symbols x_j such that $j \equiv t+1 \mod (d+1)$. Therefore, any non-delayed scheme applied to each sub-sequence will satisfy the delay constraint for the original sequence, since the last symbol in the relevant sub-sequence is x_{t-d}. Now, the sum of the Bayes envelopes corresponding to each sub-sequence is not larger than the Bayes envelope of the entire sequence, and therefore an upper bound on the total regret is at most $d+1$ times the upper bound corresponding to each sub-sequence. Since the length of each sub-sequence is about $n/(d+1)$ and the regret grows as the square root of the sequence length, the upper bound is multiplied by $\sqrt{d+1}$.

It may be somewhat surprising that a scheme that ignores most of the samples at each individual step due to sub-sampling, has a better worst-case performance than the same prediction strategy based on the entire past sequence (without the d "hidden" symbols). Even

[1] One reason for obtaining a smaller factor than with the minimax strategy is that the exponential weighting algorithm has a weighting parameter, denoted η in [6], which can be optimized taking into account the value of d. But even with the parameter value that would be selected without delay, the factor remains smaller than for the minimax strategy, namely $d+1$.

more surprising is the fact that, as shown in this paper, this simple strategy, when used in conjunction with the (non-delayed) minimax scheme, is indeed minimax for all n. Moreover, when n is a multiple of $d+1$, this result is shown for more general prediction games, in which the sequence of observations belongs to some finite alphabet A, and corresponding *actions* $b_1 b_2 \cdots b_n$ (taken from an action space B) result in instantaneous losses $\ell(b_t, x_t)$, where $\ell(\cdot,\cdot)$ denotes a non-negative function. In such games, the instantaneous loss contributions from each action-observation pair yield a cumulative loss

$$\bar{\mathcal{L}}_b(x^n) = \sum_{t=1}^{n} E_b[\ell(b_t, x_t)]$$

where the expectation accounts for a possible randomization of the strategy. In this setting, a *delayed on-line strategy* is a sequence of conditional probability distributions $p_{t+1}(\cdot|x^{t-d})$, $t = 0, 1, \cdots, n-1$, on the actions, and the regret is given by the excess loss incurred by an on-line strategy over the best constant strategy determined in hindsight, with full knowledge of x^n. In general, however, the (non-delayed) on-line minimax strategy to be used in conjunction with sub-sampling cannot be characterized as easily as Cover's scheme for the binary case with Hamming loss [8].

The delayed prediction scenario is also relevant in the logarithmic loss case, with applications to adaptive arithmetic coding. Consider a situation in which an encoder assigns a probability $p_{t+1}(x_{t+1}|x^t)$ to $x_{t+1} \in A$, based on the observed sequence x^t, in order to achieve an (ideal) code length $-\log p_{t+1}(x_{t+1}|x^t)$. Clearly, a decoder cannot start the decoding of x_{t+1} until the entire sequence $x_1 x_2 \cdots x_t$ has been decoded, which in a hardware implementation means that the process cannot be pipelined so as to reduce the number of clock cycles required by each decoding operation.[2] If, instead, the probability assigned to x_{t+1} is based only on x^{t-d}, ignoring the window of d samples $x_{t-d+1} \cdots x_{t-1} x_t$ for a suitable value of d, a pipeline can be designed. We show that the optimality of the sub-sampling strategy in a minimax sense can be extended to the logarithmic loss, provided again that $d+1$ divides n. Here, the regret (termed *pointwise redundancy* in the data compression case) is computed relative to the zero-order empirical entropy. Notice that, as in the binary case with Hamming loss, the minimax strategy without delay is well-characterized, and is given by Shtarkov's *Maximum-Likelihood* code [7].

[2] It is possible to alleviate this situation through speculative processing, but for a large alphabet A this may require an excessive amount of resources.

Since the asymptotic redundancy of this code is $(|A|-1)(\log n)/2$, the deterioration caused by the delay takes the form of a multiplicative factor of $d+1$.

The class of competing reference strategies can be extended to cover all FS predictors, as in [4] and [9], leading to the notion of *delayed FS predictability* (DFSP) of an individual sequence, which is introduced and studied in this paper. Here, rather than being constant, the competing strategy is allowed to vary according to $b_t = g(s_{t-d})$, where s_t is a state in an FS machine (FSM) with state set S, driven by a deterministic next-state function $s_{t+1} = f(s_t, x_t)$. For convenience, we assume that $s_t = s_1$ for $t \leq 1$, where s_1 is some initial state. The functions g and f, and the initial state s_1, are optimized off-line, with full knowledge of x^n, and the optimal g turns out to be deterministic, as in the non-delayed case. The delay in the state sequence reflects the constraints imposed to the on-line strategy, allowing a "fair" competition. For an infinite sequence ($n \to \infty$), the (normalized) loss incurred as $|S| \to \infty$ defines the DFSP of the sequence. For $d = 0$ and binary Hamming loss, this quantity coincides with the FS predictability of [4], which was generalized in [9] to other loss functions. The results in [9] also generalize the classical sequential decision problem [1], where the competing strategies are assumed constant. The use of FS strategies as reference models was pioneered by Ziv and Lempel in [10], in the more specific context of data compression. More general classes of reference strategies arise when these strategies are viewed as a set of generic "experts" that offer advice to the on-line decision maker [5, 11, 12, 6]. We show that, in general, the DFSP of an individual sequence is strictly larger than its FS predictability. Thus, comparing convergence rates of on-line predictors to the DFSP for different values of d is less interesting than in the single-state case, since the convergence is to a different value.

In practice, the delay applied to the prediction may not be known to the decision maker. To alleviate this problem, we will define the DFSP in a more general setting, in which each action b_t is based on full knowledge of x^{t-1}, but is "scored" relative to a number τ of future observations $x_t, x_{t+1}, \cdots, x_{t+\tau-1}$, $\tau \geq 1$. The individual loss contributions, which correspond to delays $d = 0, 1, \cdots, \tau-1$, respectively, are averaged. Specifically, we assume a loss of the form

$$\ell(b_t, x_t, x_{t+1}, \cdots, x_{t+\tau-1}) = \sum_{d=0}^{\tau-1} w_d \ell(b_t, x_{t+d}) \tag{1}$$

where w_d, $d = 0, 1, \cdots, \tau-1$, are interpreted as *weights* according to which the loss of action

b_t relative to each individual observation $x_t, x_{t+1}, \cdots, x_{t+\tau-1}$, respectively, is weighted. The expected cumulative loss takes the form

$$\bar{\mathcal{L}}_b(x^n) = \sum_{t=1}^{n-\tau+1} \sum_{d=0}^{\tau-1} w_d E_b[\ell(b_t, x_{t+d})] \,. \tag{2}$$

The setting discussed so far (excluding the first $\tau - 1$ actions) corresponds to the set of weights $w_d = 0$, $d < \tau - 1$, $w_{\tau-1} = 1$, whereas non-delayed decision making corresponds to $\tau = 1$.

In principle, it would appear that the loss in (2) leads to nothing more than a vector extension of the problem studied in [9], where the observations are vectors $\boldsymbol{X_t} \stackrel{\Delta}{=} (x_t, x_{t+1}, \cdots, x_{t+\tau-1}) \in A^\tau$, whose entries are constrained by a sliding window, and the instantaneous losses take the form

$$L(b_t, \boldsymbol{X_t}) \stackrel{\Delta}{=} \sum_{d=0}^{\tau-1} w_d \ell(b_t, x_{t+d}) \,. \tag{3}$$

However, notice that the observation $\boldsymbol{X_t}$ to which action b_t is targeted does not drive the FSM to its next state, which in turn determines b_{t+1}. Rather, the observation that determines the next state of the FSM is $\boldsymbol{X_{t-\tau+1}}$. Again, this delay reflects the fact that, in a sequential scheme, action b_t must be taken without *full* knowledge of the observations $\boldsymbol{X_{t-\tau+1}}, \boldsymbol{X_{t-\tau+2}}, \cdots, \boldsymbol{X_{t-1}}$. Nevertheless, there still exists a relation between DFSP and (non-delayed) decision making over extended alphabets, as shown in this paper. Specifically, we show that the DFSP can be achieved by non-delayed FS prediction performed on τ separate sequences of *non-overlapping* τ-vectors, for the same action space and a loss function of the form given in (3). Each such sequence results from alphabet extension on a different phase of the original sequence. It can therefore be regarded as a sub-sampling of the sequence of τ-vectors obtained by applying a *sliding window* of length τ to the original sequence. Thus, the key to this result is, again, sub-sampling.

On the other hand, the loss in (2) can be viewed as generated by a particular case of a loss function with memory, i.e., one that depends on past action-observation pairs. Such functions, which take the general form $\ell(b_{t-\tau+1}, b_{t-\tau+2}, \cdots, b_t, x_t)$ and are not covered by the classical setting of the sequential decision problem, are studied in [13]. They may capture the cost of switching from one action to another (e.g., transaction costs incurred in portfolio selection, or energy spent in control systems), or the long term effect ("memory") of an action at a given

time. The cumulative loss in (2) can be written as

$$\bar{\mathcal{L}}_b(x^n) = \sum_{t=1}^{n} \sum_{d=0}^{\tau-1} w_d E_b[\ell(b_{t-d}, x_t)] \tag{4}$$

where $\ell(b_{t-d}, x_t) \stackrel{\Delta}{=} 0$ for $t - d < 1$ and $t - d > n - \tau + 1$. Thus, the relevant loss function with memory is given by

$$\ell(b_{t-\tau+1}, b_{t-\tau+2}, \cdots, b_t, x_t) \stackrel{\Delta}{=} \sum_{d=0}^{\tau-1} w_d \ell(b_{t-d}, x_t)\,.$$

While asymptotically optimal decision schemes for loss functions with memory are devised in [13] in various settings, the proposed on-line strategies are not practical. The main difficulty resides in the fact that the loss cannot be decomposed into separate contributions from the sub-sequences occurring at each state.

In contrast, in this paper we also devise an efficient on-line algorithm for delayed decision, in the setting of competitive optimality relative to FS strategies. As in [4] and [9], the algorithm uses the Lempel-Ziv (LZ) incremental parsing rule [10]. As a universal source coding scheme, the LZ algorithm builds an implicit probabilistic model of the data [14], which can be used in on-line decision tasks other than data compression. The algorithm dynamically builds a tree, and makes decisions based on the sub-sequence of previous symbol occurrences at the current node in the traversal path. Each node corresponds to a Markovian state, given by the sequence of observations that leads from the root to the node. The decisions at each node follow on-line algorithms designed for the single-state case.[3] For example, as shown in [4], asymptotically optimal prediction follows from traversing the tree and predicting, at each step, the symbol associated with the branch most often taken at the corresponding node, up to the randomization dictated by [1] (a slightly different randomization is proposed in [4]). For more general games, it is shown in [9] that using an on-line strategy based on the LZ model, the (normalized) excess loss over the FS predictability vanishes for an arbitrary individual sequence. For delayed prediction, the asymptotic performance of the on-line scheme proposed in this paper converges to the DFSP for every individual sequence. While the connection with the non-delayed case, given by the vector extension (3) and sub-sampling, will immediately imply an LZ-based delayed prediction scheme, the proposed approach is more efficient.

[3] Since the notion of predictability applies to infinite sequences, in this context we restrict the discussion to prediction schemes that, unlike Cover's [3], are horizon-free.

The delayed prediction scenario is encountered in *adaptive prefetching* strategies for computer memory architectures. In this application, the goal is to prefetch an address from main memory into a small, faster memory ("cache") ahead of time, in order to prevent stall time by the central processing unit when accessing this address. While a first approximation to this problem is to predict the next memory reference x_t given the previous references $x_1 x_2 \cdots x_{t-1}$ (see [15]), such formulation does not address the ultimate goal, which is to have x_t already in cache *at the time it is requested*. Here, we briefly discuss the prefetching application and formalize it in terms of a Hamming loss function, for which the implementation of the above LZ-based scheme is particularly easy. It should be noted, however, that in this case the weights w_d can vary arbitrarily and are revealed to the decision maker only after the corresponding action was taken. It turns out that even under these variable conditions, the on-line scheme can still compete against *Markov* strategies (defined by an FSM for which the state at time t is given by $s_t = (x_{t-1}, \cdots, x_{t-k})$, where k is the Markov order), but fails against more general FSM strategies. Notice that a key contribution in [4] and [9] is to establish that, under mild regularity conditions, the FS predictability equals the Markov predictability, namely, that the set of competing FS machines can be reduced to the set of Markov machines. This result cannot be extended to the case of varying weights.

The remainder of this paper is organized as follows. In Section 2, we discuss minimax strategies in the single-state case, with emphasis on binary prediction. In Section 3, we introduce the notion of DFSP and investigate its properties. In Section 4, we demonstrate an LZ-based on-line algorithm for delayed decision making. Finally, in Section 5, we elaborate on the prefetching application.

2 Minimax delayed prediction

Let d denote a non-negative integer, and let $x^n = x_1 x_2 \cdots x_n$ denote a sequence over a finite alphabet A. Given a finite action space B, at each time instant t, $1 \leq t \leq n$, a (delayed) decision maker assigns a probability distribution $p_t(\cdot|x^{t-d-1})$ to an action $b_t \in B$, which ignores the last d samples and depends only on $x_1 x_2 \cdots x_{t-d-1}$, where for non-positive t, x^t denotes the null string. Each possible action b_t results in an instantaneous loss $\ell(b_t, x_t)$, where $\ell(\cdot,\cdot)$ denotes

a non-negative function. The delayed on-line strategy $\{p_t\}$ yields an expected cumulative loss

$$\bar{\mathcal{L}}_p(x^n) = \sum_{t=1}^{n} \sum_{b \in B} p_t(b|x^{t-d-1}) \ell(b, x_t) \tag{5}$$

which is compared to the Bayes envelope

$$B(x^n) = \min_{b \in B} \left\{ \sum_{t=1}^{n} \ell(b, x_t) \right\} . \tag{6}$$

The corresponding regret $R_p(x^n)$ is defined as

$$R_p(x^n) \stackrel{\Delta}{=} \bar{\mathcal{L}}_p(x^n) - B(x^n) .$$

The *minimax regret* $R_d(n)$ for delay d and sequences of length n is defined as

$$R_d(n) \stackrel{\Delta}{=} \min_p \max_{x^n \in A^n} R_p(x^n)$$

where the minimum is taken over all prediction strategies with delay d.

Part of our discussions in this section will focus on (randomized) prediction of binary sequences under Hamming loss. In this case, we can either interpret the prediction as a randomized strategy with binary actions and Hamming loss or, since the expected instantaneous loss takes the form $|x_t - p_t|$ where $p_t \stackrel{\Delta}{=} p_t(1|x^{t-d-1})$, we can view p_t as an action in the interval $[0, 1]$ under *absolute* (rather than Hamming) loss. The Bayes envelope takes the form $B(x^n) = \min(n_0(x^n), n_1(x^n))$, where $n_a(x^n)$ denotes the number of occurrences in x^n of $a \in \{0, 1\}$. For the non-delayed case ($d = 0$), the following result is due to Cover [3].

Lemma 1 *For $A = \{0, 1\}$ and Hamming loss, the non-delayed minimax regret satisfies*

$$R_0(n) = \frac{n}{2} - 2^{-n} \sum_{x^n \in A^n} B(x^n) . \tag{7}$$

Moreover, for any prediction strategy the sum of the redundancies over all sequences in A^n equals $2^n R_0(n)$, and there exists a (horizon-dependent) prediction strategy $\{p_t\}$ for which $R_p(x^n) = R_0(n)$ for all $x^n \in A^n$.

It is easy to see that the right-hand side of (7) is a lower bound on $R_0(n)$, by observing that for any prediction strategy the cumulative losses must average to $n/2$ over all sequences in A^n. A prediction strategy for which the bound is achieved with equality for *every* sequence is

demonstrated in [3] (see Appendix A). Notice that the strategy depends on the horizon n. It can readily be verified that

$$R_0(n) = 2^{-n} n \binom{n-1}{\frac{n}{2}-1} \tag{8}$$

if n is even, and $R_0(n) = R_0(n+1)$ if n is odd. Hence, using Stirling's approximation, the non-delayed minimax *normalized* regret vanishes, with its main asymptotic term taking the form $1/\sqrt{2\pi n}$.

When a possible delay d is involved in the decision, we show that the minimax regret $R_d(n)$ is achieved by a scheme that sub-samples the original sequence x^n at a rate $1/(d+1)$, and applies the non-delayed minimax strategy to each of the resulting $d+1$ sub-sequences separately. For this result to hold in cases other than binary prediction under Hamming loss, we will require n to be a multiple of $d+1$. Specifically, let $x[i]^{m_i}$ denote the sub-sequence of length $m_i = \lfloor (n+i)/(d+1) \rfloor$, $i = 0, 1, \cdots, d$, such that $x[i]_{t'} \stackrel{\Delta}{=} x_{t'(d+1)-i}$, $t' = 1, 2, \cdots, m_i$. We predict x_t by applying a non-delayed minimax strategy to $x[i_t]_1 x[i_t]_2 \cdots x[i_t]_{\lceil t/(d+1) \rceil - 1}$ with horizon m_{i_t}, where $i_t = -t \bmod (d+1)$. Notice that this scheme conforms to the delay constraint, since at the time x_t is predicted, the last symbol in the relevant sub-sequence, x_{t-d-1}, is already available. Let $R_{\rm SS}(x^n)$ denote the regret of this scheme on x^n.

Theorem 1 *Let $n_d = n \bmod (d+1)$. For $n_d = 0$ and any loss function $\ell(\cdot,\cdot)$, or for $A = \{0,1\}$, Hamming loss, and all n, the minimax regret $R_d(n)$ for delay d and sequences of length n satisfies*

$$R_d(n) = n_d R_0\left(\left\lceil \frac{n}{d+1} \right\rceil\right) + (d+1-n_d) R_0\left(\left\lfloor \frac{n}{d+1} \right\rfloor\right) . \tag{9}$$

In addition, for every $x^n \in A^n$ we have in both cases

$$R_{\rm SS}(x^n) \le R_d(n) .$$

Proof: We begin by showing that for any loss function $\ell(\cdot,\cdot)$, the worst-case regret of any given delayed on-line strategy $\{p_t\}$, applied to n-tuples with a delay d such that $d+1$ divides n, is lower-bounded by $(d+1)R_0(n_{\rm SS})$, where $n_{\rm SS} = n/(d+1)$. To this end, we will show that this lower bound applies to the expected loss under $\{p_t\}$ for a sequence x^n which is piecewise constant over blocks of length $d+1$. The key idea in the proof is to link this loss to the expected loss under an auxiliary *non-delayed* strategy for the $n_{\rm SS}$-tuple obtained by taking one

sample from each constant block in x^n. While this idea can be extended to the case $n_d \neq 0$, the manipulation of the tail of length n_d obscures the procedure, and is therefore considered separately. For any time instant t, let $\bar{t}_d \stackrel{\Delta}{=} \lceil (t+1)/(d+1) \rceil (d+1)$ denote the smallest multiple of $d+1$ not smaller than $t+1$. Since $\bar{t}_d - d - 1 \leq t$, we can define an auxiliary (non-delayed) probability assignment on the actions by

$$p'_{t+1}(b|x^t) \stackrel{\Delta}{=} \frac{1}{d+1} \sum_{i=0}^{d} p_{\bar{t}_d - i}(b|x^{\bar{t}_d - d - 1 - i}) \tag{10}$$

where, as suggested by the notation, $p'_{t+1}(b|x^t)$ depends only on x^t and is therefore an on-line strategy. Clearly, the strategy is piecewise constant over blocks of length $d+1$. Now, for each $n_{\rm SS}$-tuple $y^{n_{\rm SS}}$, let x^n denote the piecewise constant n-tuple obtained by replicating $d+1$ times each symbol in $y^{n_{\rm SS}}$, so that $x_t = y_{\lceil t/(d+1) \rceil}$, $1 \leq t \leq n$. We define a third on-line strategy, $p''_{t+1}(b|y^t)$, for $n_{\rm SS}$-tuples, by

$$p''_{t+1}(b|y^t) = p'_{(d+1)t+1}(b|x^{(d+1)t}) \,. \tag{11}$$

Given an arbitrary sequence $y^{n_{\rm SS}}$, the expected loss under $\{p_t\}$ for the corresponding (piecewise constant) n-tuple x^n satisfies, by (5),

$$\begin{aligned} \bar{\mathcal{L}}_p(x^n) &= \sum_{j=1}^{n_{\rm SS}} \sum_{i=0}^{d} \sum_{b \in B} p_{(d+1)j-i}(b|x^{(d+1)(j-1)-i}) \ell(b, y_j) \\ &= (d+1) \sum_{j=1}^{n_{\rm SS}} \sum_{b \in B} p'_{(d+1)(j-1)+1}(b|x^{(d+1)(j-1)}) \ell(b, y_j) \\ &= (d+1) \sum_{j=1}^{n_{\rm SS}} \sum_{b \in B} p''_j(b|y^{j-1}) \ell(b, y_j) = (d+1) \bar{\mathcal{L}}_{p''}(y^{n_{\rm SS}}) \end{aligned} \tag{12}$$

where the second equality follows from (10), and the third equality follows from (11). In addition, by (6) and the construction of x^n from $y^{n_{\rm SS}}$, we have in this case

$$B(x^n) = (d+1) \min_{b \in B} \left\{ \sum_{t=1}^{n_{\rm SS}} \ell(b, y_t) \right\} = (d+1) B(y^{n_{\rm SS}}) \,.$$

Therefore,

$$R_p(x^n) = (d+1) R_{p''}(y^{n_{\rm SS}}) \,. \tag{13}$$

Now, since $\{p''_t\}$ is a non-delayed on-line strategy, there exists a sequence $y^{n_{\rm SS}}$ such that

$$R_{p''}(y^{n_{\rm SS}}) \geq R_0(n_{\rm SS}) \,.$$

Thus, by (13), the corresponding piecewise constant n-tuple x^n satisfies

$$R_p(x^n) \geq (d+1)R_0(n_{\rm SS})\,. \tag{14}$$

Since $\{p_t\}$ is an arbitrary delayed on-line strategy, (14) implies

$$R_d(n) \geq (d+1)R_0(n_{\rm SS}) \tag{15}$$

as claimed.

While the same proof technique is used when n *is not* a multiple of $d+1$ to show that the right-hand side of (9) is a lower bound on $R_d(n)$ for $A = \{0,1\}$ and Hamming loss, the definition of $p'_{t+1}(b|x^t)$ needs to be adjusted for the incomplete last block of length $n_d > 0$. Specifically, while (10) holds in the range $0 \leq t < n - n_d$, for $t = n - n_d, \cdots, n-1$, we define

$$p'_{t+1}(b|x^t) \stackrel{\Delta}{=} \frac{1}{n_d} \sum_{i=0}^{n_d-1} p_{n-i}(b|x^{n-d-1-i})\,. \tag{16}$$

Since $n-d-1-i \leq n-d-1 < n-n_d$, the probability assignment on the actions still depends only on x^t. In this case, $n_{\rm SS} \stackrel{\Delta}{=} \lceil n/(d+1) \rceil$, and a piecewise constant sequence x^n is obtained from an $n_{\rm SS}$-tuple $y^{n_{\rm SS}}$ by replicating $d+1$ times each symbol in $y^{n_{\rm SS}-1}$, followed by n_d copies of $y_{n_{\rm SS}}$. Using (10) and (16), the expected loss on x^n under $\{p_t\}$ derived from (5) takes the form

$$\begin{aligned}
\bar{\mathcal{L}}_p(x^n) &= \sum_{j=1}^{n_{\rm SS}-1} \sum_{i=0}^{d} \sum_{b\in B} p_{(d+1)j-i}(b|x^{(d+1)(j-1)-i})\ell(b,y_j) + \sum_{i=0}^{n_d-1} \sum_{b\in B} p_{n-i}(b|x^{n-d-1-i})\ell(b,y_{n_{\rm SS}}) \\
&= (d+1) \sum_{j=1}^{n_{\rm SS}-1} \sum_{b\in B} p'_{(d+1)(j-1)+1}(b|x^{(d+1)(j-1)})\ell(b,y_j) \\
&+ n_d \sum_{b\in B} p'_{n-n_d+1}(b|x^{n-n_d})\ell(b,y_{n_{\rm SS}})\,.
\end{aligned} \tag{17}$$

Hence, by (11),

$$\begin{aligned}
\bar{\mathcal{L}}_p(x^n) &= (d+1) \sum_{j=1}^{n_{\rm SS}-1} \sum_{b\in B} p''_j(b|y^{j-1})\ell(b,y_j) + n_d \sum_{b\in B} p''_{n_{\rm SS}}(b|y^{n_{\rm SS}-1})\ell(b,y_{n_{\rm SS}}) \\
&= (d+1)\bar{\mathcal{L}}_{p''}(y^{n_{\rm SS}-1}) + n_d \bar{\ell}_{p''}(y_{n_{\rm SS}})
\end{aligned} \tag{18}$$

where $\bar{\ell}_{p''}(y_{n_{\rm SS}})$ denotes the expected instantaneous loss on $y_{n_{\rm SS}}$ for the strategy $\{p''_t\}$. Thus,

$$\bar{\mathcal{L}}_p(x^n) = (d+1-n_d)\bar{\mathcal{L}}_{p''}(y^{n_{\rm SS}-1}) + n_d \bar{\mathcal{L}}_{p''}(y^{n_{\rm SS}})\,. \tag{19}$$

In the special case of Hamming loss for binary sequences, which is the only one addressed when $n_d \neq 0$, notice that for any sequence x^m, there exists a symbol $a \in \{0,1\}$ which is most frequent in both x^m and x^{m-1} (possibly tied with $\bar{a}$ for one of the sequences). Therefore, it is easy to see that

$$B(x^n) = (d+1-n_d)B(y^{n_{\rm SS}-1}) + n_d B(y^{n_{\rm SS}})$$

implying, by (19),

$$R_p(x^n) = (d+1-n_d)R_{p''}(y^{n_{\rm SS}-1}) + n_d R_{p''}(y^{n_{\rm SS}})\,. \tag{20}$$

Now, define the subset of n-tuples $A_{n,d}$ by

$$A_{n,d} = \{x \in A^n \,:\, x = zy,\, z \in \{0^{d+1}, 1^{d+1}\}^{n_{\rm SS}-1},\, y \in \{0^{n_d}, 1^{n_d}\}\}$$

where zy denotes the concatenation of z and y, and for $a \in A$ and a non-negative integer i, a^i denotes the all-a i-tuple. Notice that $A_{n,d}$ is simply the set of all n-tuples obtained by replication of an $n_{\rm SS}$-tuple as shown above, and the sequences in this set are formed by juxtaposition of constant blocks of length $d+1$, followed by a tail y of length n_d. Next, we sum over $x^n \in A_{n,d}$ the regret $R_p(x^n)$. By (20), we have

$$\sum_{x \in A_{n,d}} R_p(x) = 2(d+1-n_d) \sum_{y \in A^{n_{\rm SS}-1}} R_{p''}(y) + n_d \sum_{y \in A^{n_{\rm SS}}} R_{p''}(y)\,. \tag{21}$$

By Lemma 1, the summations in the right-hand side of (21) are independent of the strategy $\{p''_t\}$, and we obtain

$$\sum_{x \in A_{n,d}} R_p(x) = 2^{n_{\rm SS}}(d+1-n_d)R_0(n_{\rm SS}-1) + 2^{n_{\rm SS}} n_d R_0(n_{\rm SS})\,. \tag{22}$$

Since the cardinality of $A_{n,d}$ is $2^{n_{\rm SS}}$, (22) implies that there exist a sequence $x^n \in A_{n,d}$ such that

$$R_p(x^n) \geq (d+1-n_d)R_0(n_{\rm SS}-1) + n_d R_0(n_{\rm SS})\,. \tag{23}$$

Together with (15), (23) implies that the right-hand side of (9) is a lower bound on the worst-case regret of any delayed on-line strategy on n-tuples with delay d for *any* value of n.

To show that the sub-sampling strategy attains the bound (9) for a general loss function, consider any sequence x^n and the corresponding sub-sequences $x[i]^{m_i}$ defined prior to Theorem 1, $0 \leq i \leq d$. Let $\bar{\mathcal{L}}^{(i)}_{\rm MM}(x[i]^{m_i})$ and $B(x[i]^{m_i})$ denote, respectively, the cumulative loss of a

(non-delayed) minimax strategy (for horizon m_i) on the sub-sequence indexed with i, and the corresponding Bayes envelope. We have

$$R_{\mathrm{SS}}(x^n) = \sum_{i=0}^{d} \bar{\mathcal{L}}_{\mathrm{MM}}^{(i)}(x[i]^{m_i}) - B(x^n) \leq \sum_{i=0}^{d} [\bar{\mathcal{L}}_{\mathrm{MM}}^{(i)}(x[i]^{m_i}) - B(x[i]^{m_i})] \leq \sum_{i=0}^{d} R_0(m_i) \qquad (24)$$

where the first inequality follows from the fact that the sum of the Bayes envelopes corresponding to each sub-sequence is not larger than the Bayes envelope of the entire sequence, and the second inequality follows from the minimax property. Since $m_i = \lfloor n/(d+1) \rfloor$ for $d+1-n_d$ sub-sampled sub-sequences, and $m_i = \lceil n/(d+1) \rceil$ for the remaining n_d sub-sequences, the proof is complete. □

Discussion. In the binary case with Hamming loss, since $R_0(n) \approx \sqrt{n/(2\pi)}$, the theorem states that $R_d(n) \approx R_0(n)\sqrt{d+1}$. As shown in Appendix A, a direct application of the minimax strategy to the delayed sequence ("hiding" a window of d symbols) yields a regret that behaves asymptotically as $R_0(n)(2d+1)$. This strategy, however, does not require prior knowledge of d. It is also shown that the asymptotic regret of the exponential weighting algorithm used in a similar fashion, in turn, behaves as $R_0(n)\sqrt{(2d+1)\pi \ln 2}$ in case d is known. If d is unknown and the algorithm employs the same weighting parameter as in the non-delayed case, the corresponding regret behaves as $R_0(n)(d+1)\sqrt{\pi \ln 2}$.

The binary case with Hamming loss allows us to establish the optimality of the sub-sampling strategy in the minimax sense for *any* value of n, not necessarily a multiple of $d+1$. Two properties contribute to this special status. First, the Bayes envelope of a piecewise constant sequence in $A_{n,d}$ is the sum of the Bayes envelopes of the sub-sampled sequences, whereas in general this is only true when n is a multiple of $d+1$. Second, a key property of Cover's minimax scheme is that the regret is uniform over all sequences $x^n \in A^n$; this property is not valid in general. Moreover, the (non-delayed) on-line minimax strategy, which is a building block for the delayed one, cannot be characterized in the general case as easily [8]. Yet, the sub-sampling strategy can still be applied to achieve vanishing normalized regret, with a possibly sub-optimal rate of convergence, in conjunction with a (non-delayed) on-line strategy that is not necessarily minimax.[4] For example, the exponential weighting algorithm of [5] and [6], when applied without delay in a context in which the "experts" are given by all possible constant

[4] In fact, in the context of FS reference strategies, we will not be concerned with the convergence rates.

strategies, yields a regret $R_{\mathrm{EW}}(x^n)$ satisfying

$$R_{\mathrm{EW}}(x^n) \le \ell_{\max}\sqrt{\frac{n\ln\beta}{2}}$$

for all $x^n \in A^n$, where β denotes the cardinality of the action space and $\ell_{\max}$ is an upper bound on the loss. Thus, the sub-sampling strategy used in conjunction with exponential weighting yields a regret which is upper bounded by $\ell_{\max}\sqrt{n(d+1)(\ln\beta)/2}$. If, in addition, we are interested in a horizon-free scheme, a modification to the algorithm is required (see [6]), at an additional cost of a constant multiplicative term in the regret.

Logarithmic loss. As discussed in Section 1, the delayed prediction scenario is also relevant in the logarithmic loss case with applications to adaptive arithmetic coding, and the proof of Theorem 1 applies almost *verbatim* in this case. Here, the decision maker assigns a probability $p_{t+1}(x_{t+1}|x^{t-d})$ to $x_{t+1} \in A$ based on the delayed sequence x^{t-d}, incurring a loss $-\log p_{t+1}(x_{t+1}|x^{t-d})$. The associated pointwise redundancy takes the form

$$R_p(x^n) = -\sum_{t=0}^{n-1}\log p_{t+1}(x_{t+1}|x^{t-d}) - n\hat{H}(x^n)$$

where $\hat{H}(x^n)$ denotes the (zero-order, normalized) *empirical entropy* of x^n, namely

$$\hat{H}(x^n) = \sum_{a\in A}\frac{n_a(x^n)}{n}\log\frac{n}{n_a(x^n)}\,.$$

Proceeding as in the proof of Theorem 1, and assuming that $d+1$ divides n, the first equality in the chain (12) translates into

$$\bar{\mathcal{L}}_p(x^n) = -\sum_{j=1}^{n/(d+1)}\sum_{i=0}^{d}\log p_{(d+1)j-i}(x_{(d+1)j-i}|x^{(d+1)(j-1)-i})\,.$$

While we cannot directly replace $\{p_t\}$ with $\{p'_t\}$ as in the second equality in (12), the key idea is to use the convexity of the log function to obtain the *inequality*

$$\bar{\mathcal{L}}_p(x^n) \ge -(d+1)\sum_{j=1}^{n/(d+1)}\log p'_{(d+1)(j-1)+1}(x_{(d+1)(j-1)+1}|x^{(d+1)(j-1)})\,.$$

Thus, proceeding as in (12),

$$\bar{\mathcal{L}}_p(x^n) \ge -(d+1)\sum_{j=1}^{n/(d+1)}\log p''_j(x_j|x^{j-1}) = (d+1)\bar{\mathcal{L}}_{p''}(y^{\frac{n}{d+1}})\,.$$

In addition, $\hat{H}(x^n) = \hat{H}(y^{\frac{n}{d+1}})$ (notice that this property does not have a counterpart when $d+1$ does not divide n). Therefore,

$$R_p(x^n) \geq (d+1)\left[\bar{\mathcal{L}}_{p''}(y^{\frac{n}{d+1}}) - \frac{n}{d+1}\hat{H}(y^{\frac{n}{d+1}})\right] = (d+1)R_{p''}(y^{\frac{n}{d+1}})\,.$$

Again, there exists a sequence $y^{\frac{n}{d+1}}$ for which $R_{p''}(y^{\frac{n}{d+1}}) \geq R_0(n/(d+1))$, where $R_0(n/(d+1))$ is the minimax pointwise redundancy without delay for sequences of length $n/(d+1)$. Consequently, the delayed minimax pointwise redundancy is at least $(d+1)R_0(n/(d+1)$. Proceeding as in the proof of Theorem 1, this bound is achieved by the sub-sampling strategy.

Notice that, as in the binary case with Hamming loss, the minimax strategy without delay is well-characterized and yields uniform pointwise redundancy. It is given by Shtarkov's *Maximum-Likelihood* (ML) code [7], which assigns to x^n a total probability

$$P^{(\mathrm{ML})}(x^n) = \frac{2^{-n\hat{H}(x^n)}}{\sum_{y^n \in A^n} 2^{-n\hat{H}(y^n)}}$$

through the sequential probability assignment

$$p^{(\mathrm{ML})}_{t+1}(x_{t+1}|x^t) = \frac{\sum_{y \in A^{n-t-1}} P^{(\mathrm{ML})}(x^{t+1}y)}{\sum_{z \in A^{n-t}} P^{(\mathrm{ML})}(x^t z)} = \frac{\sum_{y \in A^{n-t-1}} 2^{-n\hat{H}(x^{t+1}y)}}{\sum_{z \in A^{n-t}} 2^{-n\hat{H}(x^t z)}}\,.$$

Hence,

$$R_0(n) = \log\left[\sum_{y^n \in A^n} 2^{-n\hat{H}(y^n)}\right] = \frac{|A|-1}{2}\log n + O(1)$$

where the asymptotic expansion is shown in [7]. Consequently, $R_d(n) \approx (d+1)R_0(n)$.

The ML-code can be replaced by simpler, horizon-free "plug-in" assignments obtained through mixtures (see, e.g., [16]), without affecting the main asymptotic redundancy term for suitable choices of the mixture prior. In a plug-in strategy, the probability assigned to $x_{t+1} = a$ is an estimate of the probability of a if the observed sample x^t were drawn by a memoryless source, which is given by a ratio of the form $(n_a(x^t) + \gamma)/(t + |A|\gamma)$, where γ is a positive constant that depends on the mixture prior. In particular, it is shown in [7, Eq. (48)] that, for $\gamma = \frac{1}{2}$, the pointwise redundancy of *any* n-tuple differs from $R_0(n)$ by a quantity that is upper-bounded in absolute value by a constant, independent of n.

Interestingly, when any of the above asymptotically optimal schemes is used for delayed probability assignment (assigning to x_t the probability that the original scheme would have

assigned to x_{t-d}), the asymptotic worst-case redundancy in the binary case is at least $(2d+1)R_0(n)$. To see that, consider the ratio $P_d(0^{n/2}1^{n/2})/P(0^{n/2}1^{n/2})$ of the probabilities assigned to the sequence $0^{n/2}1^{n/2}$ by a scheme with and without delay, respectively. We have

$$\frac{P_d(0^{n/2}1^{n/2})}{P(0^{n/2}1^{n/2})} = \left(\frac{1}{2}\right)^d \prod_{i=0}^{d} \frac{p_{(n/2)-i+1}(1|0^{(n/2)-i})}{p_{(n/2)-i+1}(0|0^{(n/2)-i})p_{n-i+1}(1|0^{n/2}1^{(n/2)-i})} \tag{25}$$

where it is assumed that the delayed scheme assigns a probability of $\frac{1}{2}$ to each of the first d bits. It is easy to see that (25) implies that the delayed scheme assigns at least $d\log n$ more bits to the sequence than the original scheme (up to lower order terms), provided that $p_{t+1}(0|0^t) = 1-O(1/t)$ and that, for every given constant m, $p_{t-m+1}(1|0^{t/2}1^{(t/2)-m}) = \frac{1}{2}-O(1/t)$. The claim follows from observing that these conditions clearly apply to any asymptotically optimal plug-in strategy, as well as to the ML-code. Thus, the asymptotic worst-case pointwise redundancy of these schemes exceeds the optimal value obtained with the sub-sampling strategy. However, its *average* value under any i.i.d. distribution remains upper-bounded by $R_0(n)$, as stated in Section 1 for the case of Hamming loss. In addition, for the plug-in strategy, it is easy to see that the asymptotic worst-case pointwise redundancy is precisely $(2d+1)R_0(n)$. Indeed, notice that ignoring d bits in x^t results in decreasing the numerator in the probability assigned to x^{t+1} by, at most, d, whereas the denominator is decreased by d. An upper bound on the code length increase $\Delta\mathcal{L}(x^n)$ results from assuming the worst-case situation for every time instant, implying for all x^n

$$\begin{aligned}
\Delta\mathcal{L}(x^n) &\leq \sum_{i=0}^{n_0(x^n)-1} \log\left(1+\frac{d}{\gamma+i}\right) + \sum_{i=0}^{n_1(x^n)-1} \log\left(1+\frac{d}{\gamma+i}\right) - \sum_{i=1}^{d} \log(n-i+|A|\gamma) \\
&\leq \frac{2d}{\ln 2}\sum_{i=0}^{n-1}\frac{1}{\gamma+i} - d\log(n-d) = d\log n + O(1)\,.
\end{aligned}$$

Thus, the asymptotic worst-case pointwise redundancy of this scheme is $(2d+1)R_0(n)$.

Notice that, as shown in [17], the asymptotic lower bound $(|A|-1)(\log n)/2$ applies not only to the worst-case pointwise redundancy of any non-delayed probability assignment, but also to the pointwise redundancy of *most* sequences in *most* types. In contrast, the asymptotic lower bound $(d+1)(\log n)/2$ on the pointwise redundancy for delayed probability assignment on binary alphabets shown here *cannot* apply to most sequences in most types, as it would contradict the fact that for the delayed plug-in scheme with $\gamma = \frac{1}{2}$ the average under any i.i.d.

distribution is close to $\frac{1}{2}\log n$. The source of this contradiction is that, for any plug-in scheme, the possible increase in the pointwise redundancy due to the delay is not only upper-bounded by $d\log n$, but, similarly, it is lower-bounded by $-d\log n$. Thus, by [7], the asymptotic *best-case* (delayed) pointwise redundancy for $\gamma = \frac{1}{2}$ cannot be smaller than $(\frac{1}{2} - d)\log n$. Consequently, a vanishing fraction of "low redundancy" sequences in a type would not decrease the main term of the average pointwise redundancy *within the type* below $(d+1)(\log n)/2$, whereas it can be shown that the averaging distribution can be chosen so that the "exception" types have vanishing probability and do not affect the asymptotic behavior of the average.

3 Delayed FS predictability

In this section, we consider reference strategies of the form $b_t = g(s_t)$, $b_t \in B$, where s_t is a state in an FSM with state set S, driven by a next-state function $s_{t+1} = f(s_t, x_t)$, with initial state s_1. We will also extend the setting of Section 2 to loss functions of the form (1), where the weights w_d are given real numbers, and the expected cumulative loss is given by (2). The vector of weights $(w_0, w_1, \cdots, w_{\tau-1})$ is denoted by $\boldsymbol{w}$, and the setting of Section 2 corresponds to $\boldsymbol{w} = (0, 0, \cdots, 0, 1)$ (here, however, we exclude the first $\tau - 1$ actions). Clearly, the best reference strategy g for given f and s_1 achieves, over $n - \tau + 1$ actions, the (normalized) loss

$$\bar{\mu}^{\boldsymbol{w}}_{f,s_1}(x^n) = \sum_{s\in S} p_{x^{n-\tau+1}}(s) \min_{b\in B}\left\{\sum_{d=0}^{\tau-1} w_d \sum_{u\in A^d}\sum_{a\in A} p_{x^{n-\tau+d+1}}(ua|s)\ell(b,a)\right\} \tag{26}$$

where $p_{x^j}(s)$ denotes the frequency of occurrence of $s \in S$ in the state sequence $s_1 s_2 \cdots s_j$ and, likewise, the conditional empirical probability $p_{x^j}(ua|s)$ (based on x^j) is defined as the frequency with which the $(d+1)$-vector $x_t x_{t+1} \cdots x_{t+d}$ is ua, given that $s_t = s$, $0 < t \le j - d$. Thus, for an infinite sequence of observations $x^\infty = x_1 x_2 \cdots$, the asymptotic performance achievable by the best FS strategy determined in hindsight is given by

$$\bar{\mu}^{\boldsymbol{w}}(x^\infty) = \lim_{\Omega\to\infty} \limsup_{n\to\infty} \min_{s_1\in S} \min_{f:|S|=\Omega} \bar{\mu}^{(\boldsymbol{w})}_{f,s_1}(x^n)\,. \tag{27}$$

We define this value as the *delayed* FS predictability of x^∞ for the vector $\boldsymbol{w}$. Notice that, for $\tau = 1$, the DFSP coincides with the (generalized) FS predictability of [9] for the loss function $w_0\ell(\cdot,\cdot)$.

In the remainder of this section, we establish some properties of the DFSP. The first property relates the DFSP to a non-delayed measure of predictability through sub-sampling. We then show that, as in the non-delayed case, Markov machines achieve the same asymptotic performance as the broader set of general FSM's, when the number of states grows. Finally, we show that the DFSP is a *proper* generalization of the usual FS predictability of [9]. These properties are applied in Section 4 to the design of on-line algorithms that achieve the DFSP.

Given an infinite sequence x^∞ over A, let $\boldsymbol{X}[i]^\infty$, $0 \le i < \tau$, denote the infinite sequence over A^τ such that $\boldsymbol{X}[i]_t = (x_{t\tau-i}, x_{t\tau-i+1}, \cdots, x_{(t+1)\tau-i-1})$, $t = 1, 2, \cdots$. Notice that the sequences $\boldsymbol{X}[i]^\infty$, formed by *non-overlapping* blocks over x^∞ taken at a "phase" given by i, are the sub-sequences resulting from sub-sampling the sequence of τ-vectors obtained by applying a sliding window of length τ to x^∞. For an observation space A^τ and an action space B, we consider *non-delayed* FS prediction of each sub-sequence $\boldsymbol{X}[i]^\infty$, $0 \le i < \tau$, under the loss function defined in (3), namely

$$L(b, \boldsymbol{Y}) = \sum_{d=0}^{\tau-1} w_d \ell(b, y_d) \tag{28}$$

where $\boldsymbol{Y} = (y_0, y_1, \cdots, y_{\tau-1}) \in A^\tau$ and $b \in B$. In this setting, a different FSM, with next-state function $f(i)$ and initial state $s_1(i)$, acts on each sub-sequence $\boldsymbol{X}[i]^\infty$, and is optimized separately. For a prefix x^n of x^∞, this prediction accumulates τ independent losses over the corresponding sub-sequences $\boldsymbol{X}[i]^{n(i)}$, where $n(i) \stackrel{\Delta}{=} \lfloor (n-\tau+1+i)/\tau \rfloor$ is the length of the longest prefix of $\boldsymbol{X}[i]^\infty$ (over A^τ) contained in x^n. These losses are added together, with $\bar{\nu}^{\boldsymbol{w}}_{f(i),s_1(i)}(\boldsymbol{X}[i]^{n(i)})$ denoting the (normalized) loss achieved over $\boldsymbol{X}[i]^{n(i)}$ by the best (non-delayed) strategy for the FSM determined by $f(i)$ and $s_1(i)$. A "sub-sampled predictability" of x^∞ is defined by

$$\bar{\rho}^{\boldsymbol{w}}(x^\infty) \stackrel{\Delta}{=} \lim_{\Omega\to\infty} \limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{\tau-1} n(i) \min_{s_1(i)\in S(i)} \min_{f(i):|S(i)|=\Omega} \bar{\nu}^{(\boldsymbol{w})}_{f(i),s_1(i)}(\boldsymbol{X}[i]^{n(i)}) \,. \tag{29}$$

Notice that if the summation in i can be interchanged with the limit superior in n (e.g., in case the limit in n exists for all the sub-sequences), then $\bar{\rho}^{\boldsymbol{w}}(x^\infty)$ is just the average of the (non-delayed) FS predictabilities $\bar{\nu}^{\boldsymbol{w}}(\boldsymbol{X}[i]^\infty)$ of $\boldsymbol{X}[i]^\infty$, $0 \le i < \tau$, as defined in [9]. In general, however, this average is only an upper bound on the new predictability measure (29) which, as shown in Theorem 2 below, coincides with the DFSP.

Theorem 2 *For any positive integer τ, any vector $\boldsymbol{w}$, and any infinite sequence x^∞, we have*

$$\bar{\mu}^{\boldsymbol{w}}(x^\infty) = \bar{\rho}^{\boldsymbol{w}}(x^\infty)\,. \tag{30}$$

Theorem 2 tells us that in order to achieve the DFSP, it is possible to consider τ separate sub-sampled sub-sequences over the extended alphabet, and apply known techniques for non-delayed decision making. By emphasizing, again, the optimality of sub-sampling, this result immediately implies an *on-line* algorithm for approaching the DFSP: It suffices to run in parallel τ on-line schemes for non-delayed decision over the extended alphabet, one for each phase of the sequence. On-line algorithms for approaching the DFSP are further discussed in Section 4.

Proof of Theorem 2: Consider an FSM with next-state function f over a set of states S, which is started at state s_1 and is driven by the symbols in A. By abuse of notation, given $a \in A$, $s \in S$, and a string u over A, we recursively define $f(s,au) = f(f(s,a),u)$. We create a refinement of the FSM by splitting each state $s \in S$ into τ states denoted $s^{(0)}, s^{(1)}, \cdots, s^{(\tau-1)}$, and by defining a new next-state function, f_τ, such that for any $a \in A$ and any d, $0 \le d < \tau$, $f_\tau(s^{(d)},a) = s'^{(d')}$, where $s' = f(s,a)$ and $d' \equiv d-1 \bmod \tau$. The initial state of the refined machine is selected to be $s_1^{(\tau-1)}$, so that state $s^{(i)}$ can only occur at times $t = j\tau - i$, namely, at the beginning of block $\boldsymbol{X}[i]_j$, $j = 1,2,\ldots$. By the refinement property, for every prefix x^n of x^∞ we have

$$\bar{\mu}^{\boldsymbol{w}}_{f_\tau,s_1^{(\tau-1)}}(x^n) \le \bar{\mu}^{\boldsymbol{w}}_{f,s_1}(x^n)\,. \tag{31}$$

Clearly, by (26), (28), and the fact that for any string u and any length j, $\sum_{u' \in A^j} p_{x^n}(uu'|s) = p_{x^{n-j}}(u|s)$, we have

$$\begin{aligned}
\bar{\mu}^{\boldsymbol{w}}_{f_\tau,s_1^{(\tau-1)}}(x^n) &= \sum_{s\in S}\sum_{i=0}^{\tau-1} p_{x^{n-\tau+1}}(s^{(i)}) \min_{b\in B}\left\{\sum_{d=0}^{\tau-1} w_d \sum_{u\in A^d}\sum_{a\in A}\sum_{u'\in A^{\tau-d-1}} p_{x^n}(uau'|s^{(i)})\ell(b,a)\right\} \\
&= \sum_{s\in S}\sum_{i=0}^{\tau-1} p_{x^{n-\tau+1}}(s^{(i)}) \min_{b\in B}\left\{\sum_{\boldsymbol{Y}\in A^\tau} p_{x^n}(\boldsymbol{Y}|s^{(i)})L(b,\boldsymbol{Y})\right\}\,.
\end{aligned} \tag{32}$$

Now, we define yet another FSM over S, whose next-state function F_τ is driven by the symbols in A^τ, according to $F_\tau(s,\boldsymbol{Y}) = f(s,y_1y_2\cdots y_\tau)$, where $\boldsymbol{Y} = (y_1,y_2,\cdots,y_\tau)$, $y_j \in A$, $j = 1,2,\cdots,\tau$. Clearly, each occurrence of $s^{(i)}$ in $x^{n-\tau+1}$ corresponds to an occurrence of the state s (driven by F_τ), in $\boldsymbol{X}[i]^{n(i)}$. Therefore, (32) implies

$$\bar{\mu}^{\boldsymbol{w}}_{f_\tau,s_1^{(\tau-1)}}(x^n) = \frac{1}{n-\tau+1}\sum_{i=0}^{\tau-1} n(i) \sum_{s\in S} p_{\boldsymbol{X}[i]^{n(i)}}(s) \min_{b\in B}\left\{\sum_{\boldsymbol{Y}\in A^\tau} p_{\boldsymbol{X}[i]^{n(i)}}(\boldsymbol{Y}|s)L(b,\boldsymbol{Y})\right\}$$

$$= \frac{1}{n-\tau+1}\sum_{i=0}^{\tau-1} n(i)\,\bar{\nu}^{\boldsymbol{w}}_{F_\tau,s_{\tau-i}}(\boldsymbol{X}[i]^{n(i)})$$

and, using (31), minimizing over f and s_1, taking the limit superior on n, and letting $|S| \to \infty$, we obtain

$$\bar{\mu}^{\boldsymbol{w}}(x^\infty) \geq \bar{\rho}^{\boldsymbol{w}}(x^\infty)\,.$$

To prove the opposite inequality, it is convenient to invoke the asymptotic equivalence between Markov and FSM predictability shown in [9, Theorem 2] for the non-delayed case. Notice that the conditions for this equivalence clearly hold in the finite matrix games considered in this paper. Therefore, for any Markov order k, and any set of FSM's with next-state functions $f(i)$ and initial states $s(i)$ over a set of Ω states, we have

$$\frac{1}{n}\sum_{i=0}^{\tau-1} n(i)\,\bar{\nu}^{(\boldsymbol{w})}_{f(i),s_1(i)}(\boldsymbol{X}[i]^{n(i)}) \geq \frac{1}{n}\sum_{i=0}^{\tau-1} n(i)\,\bar{\nu}^{\boldsymbol{w}}_{\mathcal{M}_k}(\boldsymbol{X}[i]^{n(i)}) - \delta(k,\Omega) \tag{33}$$

where $\mathcal{M}_k$ denotes the next-state function of a Markov machine of order k with an arbitrary initial state, and the function $\delta(k,\Omega)$ vanishes as k tends to infinity, provided that $\Omega = o(2^k)$. The sum in the right-hand side of (33) accumulates the losses achieved on all "phases" of x^∞ by separate Markovian machines of order k, driven by the symbols in A^τ, with the loss function $L(\boldsymbol{Y}, b)$. By the Markov property, the same loss can be achieved by a single FSM driven by the symbols in A, whose state is given by the last $k\tau$ symbols and the phase i, with the loss function of (1). Notice that the state space $S_{k,\tau}$ of this FSM has $\tau|A|^{k\tau}$ states (and is not Markovian). Therefore,

$$\frac{1}{n}\sum_{i=0}^{\tau-1} n(i) \min_{s_1(i)\in S(i)} \min_{f(i):|S(i)|=\Omega} \bar{\nu}^{(\boldsymbol{w})}_{f(i),s_1(i)}(\boldsymbol{X}[i]^{n(i)}) \geq \min_{s_1\in S_{k,\tau}} \min_{f:|S_{k,\tau}|=\tau|A|^{k\tau}} \bar{\mu}^{\boldsymbol{w}}_{f,s_1}(x^n) - \delta(k,\Omega)\,.$$

The result follows from taking the limit superior as $n \to \infty$ and then letting Ω (and, therefore, k) tend to infinity. □

While the proof of Theorem 2 also implies that the class of Markov machines which are also equipped with information on the phase i of the sequence is as powerful as the entire FSM class in the sense of achieving the DFSP, Theorem 3 below states a stronger property. Specifically, it is shown that, just as in the non-delayed case, the (delayed) Markov predictability is equivalent to the DFSP. Thus, the phase information is asymptotically inconsequential.[5]

[5]It should be noted, however, that the weaker result implied by Theorem 2 does not appear to be useful in the proof of Theorem 3.

Theorem 3 *For any positive integer τ, any vector $\boldsymbol{w}$, and any infinite sequence x^∞, we have*

$$\bar{\mu}^{\boldsymbol{w}}(x^\infty) = \lim_{k\to\infty} \limsup_{n\to\infty} \bar{\mu}^{\boldsymbol{w}}_{\mathcal{M}_k}(x^n)\,.$$

Proof. The theorem results from establishing, for every next-state function f over a state space S, every initial state s_1, and every non-negative integer k, the inequality

$$\bar{\mu}^{\boldsymbol{w}}_{\mathcal{M}_k}(x^n) - \bar{\mu}^{\boldsymbol{w}}_{f,s_1}(x^n) \le W\ell_{\max}\sqrt{\frac{2\ln|S|}{1+\left\lfloor\frac{k}{\tau+1}\right\rfloor}} \tag{34}$$

and letting $n\to\infty$ and $k\to\infty$, where $W \stackrel{\Delta}{=} \sum_{i=0}^{\tau-1} w_d$ and we recall that $\ell_{\max}$ denotes an upper bound on the loss. Following the method for proving the counterpart of this inequality in [9, Theorem 2], we upper-bound the left-hand side with an average of differences of the form $\bar{\mu}^{\boldsymbol{w}}_{\mathcal{M}_j}(x^n) - \bar{\mu}^{\boldsymbol{w}}_{f_j,s_1}(x^n)$, where f_j denotes a common refinement of f and $\mathcal{M}_j$, and $j\le k$. However, unlike in [9], we let the integers j take the form $j=m\tau$, where $0\le m\le\lfloor k/\tau\rfloor$. These differences are bounded, in turn, as shown in Lemma 2 below (which replaces Equation (A.3) of [9]). Notice that the auxiliary empirical conditional entropies in Lemma 2 correspond to distributions on τ-tuples, rather than on single letters as in [9].

Lemma 2 *Let a refinement of a Markov machine of order j have state space S next-state function f, and initial state s_1. Let $\hat{H}(X^\tau|X^j)$ and $\hat{H}(X^\tau|\mathcal{S})$ denote the conditional entropies of the empirical distributions on τ-tuples conditioned on the Markov and the refined machine, respectively (with frequencies dictated by x^n). Then,*

$$\bar{\mu}^{\boldsymbol{w}}_{\mathcal{M}_j}(x^n) - \bar{\mu}^{\boldsymbol{w}}_{f,s_1}(x^n) \le W\ell_{\max}\sqrt{2(\ln 2)[\hat{H}(X^\tau|X^j) - \hat{H}(X^\tau|\mathcal{S})]}\,.$$

Lemma 2 is proved in Appendix B. The rest of the proof of Theorem 3 is omitted, since it proceeds as in [9], except that here the chain rule of conditional entropies is used on τ-tuples. □

In case $\boldsymbol{w}$ is the τ-vector $(0,0,\cdots,0,1)$, we will denote $\bar{\mu}^{\boldsymbol{w}}_{f,s_1}(x^n) \stackrel{\Delta}{=} \bar{\mu}^{(\tau-1)}_{f,s_1}(x^n)$. The DFSP of x^∞, which we will denote $\bar{\mu}^{(\tau-1)}(x^\infty)$, gives the minimum loss per sample incurred by any FS decision maker that acts $\tau-1$ steps in advance (or observes a sequence with a delay $\tau-1$). We will refer to $\bar{\mu}^{(\tau-1)}(x^\infty)$ as the DFSP *of order* $\tau-1$. It is easy to see that for any sequence x^n, any FSM defined by f and s_1, any positive integer τ, and any vector of weights $\boldsymbol{w}$,

$$\bar{\mu}^{\boldsymbol{w}}_{f,s_1}(x^n) \ge \sum_{d=0}^{\tau-1} w_d \bar{\mu}^{(d)}_{f,s_1}(x^n)\,.$$

Thus, the achievable loss for $\boldsymbol{w}$ is lower-bounded by a linear combination of delayed predictabilities of orders $0, 1, \cdots, \tau - 1$.

As noted in Section 2, the performance achieved by a *single-state* machine is independent of the delay. However, when general FSM's are considered, the concept of delayed predictability is a proper generalization of the usual predictability. The following theorem states that not only the DFSP cannot decrease with the order, but there indeed exist sequences for which "longer delay" decision making is sub-optimal.

Theorem 4

a. For any infinite sequence of observations x^∞, $\mu^{(\tau-1)}(x^\infty) \leq \mu^{(\tau'-1)}(x^\infty)$ *for all* $\tau' \geq \tau > 0$.

b. Assume that there exist $a_1, a_2 \in A$ *such that*

$$\ell(\arg\min_{b\in B}\ell(b, a_1), a_2) > \min_{b\in B}\ell(b, a_2)$$

(namely, the loss function is non-trivial, in the sense that there is no single action that dominates all other actions regardless of the observation). Then, for any $\tau > 1$, *there exist infinite sequences* x^∞ *for which* $\bar{\mu}^{(\tau-1)}(x^\infty) > \bar{\mu}^{(0)}(x^\infty)$.

Proof. The first part of the theorem is straightforward, since an FSM that incurs a loss $\ell(g(s_{t-\tau'+1}), x_t)$ on x_t for some function g of the state at time $t - \tau' + 1$, cannot degrade its performance if g is allowed to depend also on $x_{t-\tau'+1}, \cdots, x_{t-\tau}$.

As for the second part, it suffices to show that the *strict* inequality holds with probability one for sequences emitted by some ergodic first-order Markov (probabilistic) source $\mathcal{S}$. By the assumption on the loss function, there exist such sources, with conditional distributions $p(\cdot|\cdot)$, that for some $\epsilon > 0$ satisfy the inequality

$$\sum_{v\in A} p(v) \min_{b\in B}\left\{\sum_{u\in A^{\tau-1}}\sum_{a\in A} p(ua|v)\ell(b, a)\right\} \geq \sum_{v\in A} p(v) \sum_{u\in A^{\tau-1}} \min_{b\in B}\left\{\sum_{a\in A} p(ua|v)\ell(b, a)\right\} + \epsilon \quad (35)$$

where $\{p(v),\ v \in A\}$ denotes the steady state distribution derived from $p(\cdot|\cdot)$. Fix k and consider the Markov machine $\mathcal{M}_k$ with an arbitrary initial state. By Birkhoff's ergodic theorem, for sequences x^n emitted by $\mathcal{S}$, the $k + \tau$ empirical joint distribution tends to the true distribution almost surely as $n \to \infty$. Hence, by (26) and the continuity of the (delayed) Bayes envelope,

$$\limsup_{n\to\infty} \bar{\mu}^{(\tau-1)}_{\mathcal{M}_k}(x^n) = \sum_{v\in A^k} p(v) \min_{b\in B}\left\{\sum_{u\in A^{\tau-1}}\sum_{a\in A} p(ua|v)\ell(b, a)\right\} \stackrel{\Delta}{=} \bar{\mu}^{(\tau-1)}_k(\mathcal{S})$$

almost surely. Now, the idea is to prove the theorem in the probabilistic setting. Specifically, since $\mathcal{S}$ is Markov of order 1, by (35), we have

$$\begin{aligned}
\bar{\mu}_k^{(\tau-1)}(\mathcal{S}) &= \bar{\mu}_1^{(\tau-1)}(\mathcal{S}) = \sum_{v\in A} p(v) \min_{b\in B} \left\{ \sum_{u\in A^{\tau-1}} \sum_{a\in A} p(ua|v)\ell(b,a) \right\} \\
&\geq \sum_{v\in A} p(v) \sum_{u\in A^{\tau-1}} \min_{b\in B} \left\{ \sum_{a\in A} p(ua|v)\ell(b,a) \right\} + \epsilon \\
&= \sum_{v\in A} p(v) \sum_{u\in A^{\tau-1}} p(u|v) \min_{b\in B} \left\{ \sum_{a\in A} p(a|vu)\ell(b,a) \right\} + \epsilon \\
&= \sum_{u'\in A^{\tau}} p(u') \min_{b\in B} \left\{ \sum_{a\in A} p(a|u')\ell(b,a) \right\} + \epsilon \\
&= \sum_{v\in A} p(v) \min_{b\in B} \left\{ \sum_{a\in A} p(a|v)\ell(b,a) \right\} + \epsilon \\
&= \sum_{u\in A^{k}} p(u) \min_{b\in B} \left\{ \sum_{a\in A} p(a|u)\ell(b,a) \right\} + \epsilon = \bar{\mu}_k^{(0)}(\mathcal{S}) + \epsilon \,. \qquad (36)
\end{aligned}$$

Again, by the ergodic theorem,

$$\limsup_{n\to\infty} \bar{\mu}_{\mathcal{M}_k}^{(0)}(x^n) = \bar{\mu}_k^{(0)}(\mathcal{S})$$

almost surely, and therefore, by (36),

$$\limsup_{n\to\infty} \bar{\mu}_{\mathcal{M}_k}^{(\tau-1)}(x^n) \geq \limsup_{n\to\infty} \bar{\mu}_{\mathcal{M}_k}^{(0)}(x^n) + \epsilon \qquad (37)$$

with $\mathcal{S}$-probability 1. Since the (countable) intersection of probability-1 sets also has probability 1, we can let $k \to \infty$ in (37). Finally, since $\epsilon > 0$, by Theorem 3, we conclude that

$$\bar{\mu}^{(\tau-1)}(x^\infty) > \bar{\mu}^{(0)}(x^\infty)$$

with $\mathcal{S}$-probability 1. □

4 Delayed decision making via incremental parsing

In this section, we propose a sequential algorithm that, for an arbitrary sequence of observations x^n, incurs a loss $\bar{\mathcal{L}}_b(x^n)$, as defined in (2), which approaches the DFSP for any given loss function $\ell(\cdot,\cdot)$ and weight τ-vector $\boldsymbol{w}$. By Theorem 2, one possible approach to achieve asymptotically

optimal delayed decision making is to reduce the problem to the non-delayed case studied in [9], through sub-sampling. Specifically, we can grow τ LZ trees in parallel over the extended alphabet A^τ, one for each sequence $\boldsymbol{X}[i]$, $0 \leq i < \tau$, and use the LZ-based sequential decision making scheme of [9] for the loss function (3). Each tree yields a sub-sampled sequence of decisions, at the corresponding phase i, and the compound loss converges to the DFSP. While this approach is plausible, a more direct application of the incremental parsing rule to delayed decision making avoids the costs associated with alphabet extension (especially in terms of memory usage), as shown next.

Following [4], we will first derive a strategy that competes successfully against any *given* Markov machine. By Theorem 3, a Markov machine of sufficiently large order can perform as well as any given FSM. We will then take advantage of the growing Markov order induced by the LZ incremental parsing rule to design the desired sequential algorithm. For simplicity, we will assume $|A| = |B|$, so that a one-to-one mapping between observations and actions can be defined. By abuse of notation, we will denote $b = a$ for corresponding values $b \in B$ and $a \in A$ under this mapping. Moreover, we will restrict our analysis to the Hamming loss, namely,

$$\ell(b, a) \stackrel{\Delta}{=} \begin{cases} 0 & \text{if } b = a \\ 1 & \text{otherwise.} \end{cases} \tag{38}$$

While our results carry over to general loss functions, the Hamming loss facilitates an efficient implementation of the sequential decision scheme. Moreover, the prefetching application to be discussed in Section 5 conforms to this restriction (except that the weights can vary arbitrarily and are only revealed to the decision maker after the corresponding action was taken).

4.1 Sequential strategies for Markov models

For non-delayed decision making, a sequential scheme that performs essentially as well as the best *constant* strategy determined in hindsight can readily be extended to compete against the best Markov predictor of a *given* order k, by using it on the sub-sequences of observations following each k-tuple. Specifically, for a sub-sequence of length $n(s)$ occurring at state s, an $O(\sqrt{n(s)})$ excess loss with respect to the best constant strategy can be sequentially achieved. Therefore, integrating this result over the state space S of the competing Markov machine

through Jensen's inequality, we obtain an overall (normalized) regret (with respect to the best Markov strategy) which is $O(\sqrt{|S|/n})$ for a sequence of length n (see [4]).

It is possible to extend these considerations to delayed prediction by proceeding as in Theorem 2, creating τ separate sequences of observations over A^τ, which in turn are divided according to the Markov state in which the observation vector occurred. The actions still belong to B, and the loss function is given by (3). For each phase and state, a horizon-free decision scheme with $O(\sqrt{n})$ regret (such as the exponential weighting scheme discussed at the end of Section 2) is applied, and thus the dependency of the regret on $|S|$ and τ is given by an $O(\sqrt{\tau|S|n})$ term. Consequently, there is a regret cost for both the delay and the number of states. However, when dealing with sequences such that two consecutive occurrences of a given state are always at least τ time instants apart (namely, $s_t = s$ implies $s_{t+i} \neq s$ for $i = 1, 2, \cdots, \tau - 1$), the delay cost can be avoided. The reason is that, for such sequences, the vector of observations $\boldsymbol{X_t}$ occurring at state $s_t = s$ (and the corresponding loss $L(b_t, \boldsymbol{X_t})$ for action b_t) is already available at the next occurrence of state s. Consequently, it is not necessary to consider τ separate phases. This is a key observation in the context of approaching the Markov predictability, as it will become clear that the assumption is not restrictive due to the use of the incremental parsing rule to build a machine of growing Markov order.

For a sub-sequence of observations $\boldsymbol{X_{t_1}}, \boldsymbol{X_{t_2}}, \cdots, \boldsymbol{X_{t_{i(s)}}}$ over A^τ occurring at state s, all of which are assumed to be available at the time of the next visit to s, the exponential weighting algorithm assigns to action $b \in B$ a probability

$$P(b|\boldsymbol{X_{t_1}}, \boldsymbol{X_{t_2}}, \cdots, \boldsymbol{X_{t_{i(s)}}}) = \frac{e^{-\eta \mathcal{L}_b(s,i)}}{\sum_{b' \in B} e^{-\eta \mathcal{L}_{b'}(s,i)}} \tag{39}$$

where $\mathcal{L}_b(s,i)$ is the cumulative loss of action b for the sub-sequence, namely

$$\mathcal{L}_b(s,i) \stackrel{\Delta}{=} \sum_{j=1}^{i(s)} L(b, \boldsymbol{X_{t_j}}) \tag{40}$$

and η is a constant whose optimal value depends on the length $n(s)$ of the sub-sequence. Since $n(s)$ depends on the sequence x^n, it cannot be assumed known even in cases where the horizon n is known. To address this problem, it is proposed in [6] to divide time into exponentially growing super-segments, and to apply the above algorithm to each super-segment independently, optimizing η for the corresponding length. Notice that the cumulative loss $\mathcal{L}_b(s, n_i)$, where n_i is

the length of the i-th super-segment, is reset before starting super-segment $i+1$. The normalized regret is bounded as in the horizon-dependent case, but with a larger constant [6].

4.2 Delayed decision algorithm

In order to compete against any FSM, we will rely on the incremental parsing rule of [10] to increase the Markov order at a suitable rate. Based on this rule, the decision algorithm will grow the same tree as in the data compression application, but the count updates will differ from those specified in [4] for binary (non-delayed) prediction. The branches in the tree represent observations, and a node represents a Markovian state, through the unique path from the root (the reader is referred to [10] and [4] for further details). In addition, for each node $\mathcal{N}$, a count $c_b(\mathcal{N})$ is associated with each action $b \in B$. As shown in Equation (41) below, this count stores a (non-negative) difference between a node-dependent reference value and the cumulative loss that would be incurred by a constant strategy that uses action b over the sub-sequence of observations occurring at the state represented by $\mathcal{N}$. The counts $c_b(\mathcal{N})$ are initialized to 0, and are reset occasionally to account for the exponentially growing super-segments discussed in Section 4.1. To this end, a counter $n(\mathcal{N})$ registers the number of visits to $\mathcal{N}$, and determines a super-segment index $m(\mathcal{N})$.

The proposed sequential strategy is described through a pointer that at time $t-1$ is pointing to node $\mathcal{N}_{j-1}$ at level $j-1$ of the tree, after having pointed to each node in the path $\mathcal{N}_0 \cdots \mathcal{N}_{j-2}\,\mathcal{N}_{j-1}$ from the root $\mathcal{N}_0$ (initially, $j=1$). If $j < \tau$, we also keep track of additional nodes visited before the last return to the root, which are denoted $\mathcal{N}_{-1}, \mathcal{N}_{-2}, \cdots, \mathcal{N}_{j-\tau}$ (from the most recent to the most remote), so as to complete a history of length τ. Thus, at a given time, a given node may be labeled by multiple indexes, only one of which can be non-negative (the level in the tree). At that point, an observation $x_{t-1} \in A$ occurs. The strategy proceeds as follows:

a. For $d = 0$ to $\tau - 1$, increment by w_d the count $c_b(\mathcal{N}_{j-1-d})$, for the action $b = x_{t-1}$.

b. Traverse the tree in the direction of x_{t-1}, moving the pointer to $\mathcal{N}_j$. If the branch does not exist in the tree, add it and reset the pointer to the root $\mathcal{N}_0$ ($j = 0$); in this process, the node previously pointed to is re-labeled $\mathcal{N}_{-1}$, and a history of length τ is maintained.

c. Draw action b_t according to the distribution

$$P(b|\mathcal{N}_j) = \frac{e^{\eta(m(\mathcal{N}_j))c_b(\mathcal{N}_j)}}{\sum_{b' \in B} e^{\eta(m(\mathcal{N}_j))c_{b'}(\mathcal{N}_j)}}$$

where $\eta(m(\mathcal{N}_j))$ is the parameter in the exponential weighting algorithm associated with the super-segment index $m(\mathcal{N}_j)$.

d. Update $n(\mathcal{N}_j)$; if the update indicates the beginning of a new super-segment, update $m(\mathcal{N}_j)$ and reset all counts c_b associated with $\mathcal{N}_j$.

The counts $c_b(\mathcal{N})$ in the above procedure differ from those used in [4] in that each observation x_t generates up to τ updates into the past. The parsing rule, however, which is determined by the return to the root, is the same as in [10]. Each count updated in Step a. corresponds to the only action b that, if executed at time $t-1-d$, would not have contributed to the loss component $w_d \ell(b_{t-1-d}, x_{t-1})$ (the other actions would have contributed w_d), $0 \le d < \tau$. Thus, with $\boldsymbol{T_j}(t)$ denoting the set of time instants $t_1, t_2, \cdots, t_{n_t(\mathcal{N}_j)} < t$ such that the decision b_{t_i} was made at node $\mathcal{N}_j$, $1 \le i \le n_t(\mathcal{N}_j)$, if $t_{n_t(\mathcal{N}_j)} + \tau - 1 < t$ then for every action $b \in B$ we have, at time t,

$$\begin{aligned} c_b(\mathcal{N}_j) &= \sum_{i \in \boldsymbol{T_t}(t)} \sum_{d=0}^{\tau-1} w_d[1 - \ell(b, x_{i+d})] = W n_t(\mathcal{N}_j) - \sum_{i \in \boldsymbol{T_j}(t)} L(b, \boldsymbol{X_i}) \\ &= W n_t(\mathcal{N}_j) - \mathcal{L}_b(\mathcal{N}_j, n_t(\mathcal{N}_j)) \end{aligned} \tag{41}$$

where the second equality in the chain follows from (3), and the third equality follows from (40), with the node $\mathcal{N}_j$ playing the role of a state. Notice that the condition $t_{n_t(\mathcal{N}_j)} + \tau - 1 < t$ guarantees that all previous instantaneous losses $L(b, \boldsymbol{X_i})$, $i \in \boldsymbol{T_j}(t)$, have been added to $c_b(\mathcal{N}_j)$; in particular, no "edge effects" result from the return to the root, since the nodes labeled with negative indexes ensure the availability of the complete history of length τ. Thus, under this condition, by (39) and (41), Step c. of the algorithm implements the exponential weighting algorithm for the subsequence $\{x_i\}$, $i \in \boldsymbol{T_j}(n)$.

Theorem 5 *Let $\bar{\mathcal{L}}_{\mathrm{LZ}}^{\boldsymbol{w}}(x^n)$ denote the (expected) loss incurred by the above on-line strategy over a sequence x^n, for a weight vector $\boldsymbol{w}$. Then, for any Markov order k,*

$$\limsup_{n \to \infty} \frac{1}{n} \left[\bar{\mathcal{L}}_{\mathrm{LZ}}^{\boldsymbol{w}}(x^n) - \bar{\mu}_{\mathcal{M}_k}^{\boldsymbol{w}}(x^n) \right] \le 0 \,.$$

Proof. Let $c(x^n)$ denote the number of phrases in the incremental parsing of x^n, and let $\kappa = \max(k, \tau - 1)$. As in [4], the proof uses the fact that there are at most $\kappa \cdot c(x^n)$ observations at nodes of level j, $j \leq \kappa$, and therefore the loss contributed by actions decided at these nodes is at most $\kappa W \ell_{\max} c(x^n)$.

Now, consider the loss due to actions decided at nodes of level j, $j > \kappa$. Notice that the evolution of the tree guarantees that when an action b_t is decided at a node $\mathcal{N}_j$, then the time $t_{n_t(\mathcal{N}_j)}$ at which the last action was decided at $\mathcal{N}_j$ satisfies $t_{n_t(\mathcal{N}_j)} < t - j$. Since $j > \tau - 1$, the condition $t_{n_t(\mathcal{N}_j)} < t - \tau + 1$ is satisfied, and Step d. of the algorithm indeed implements the exponential weighting algorithm for the subsequence occurring at $\mathcal{N}_j$. Hence, by the discussion in Section 4.1, the difference between the cumulative loss for actions decided at $\mathcal{N}_j$ and the loss that would be obtained with the best fixed strategy determined in hindsight for this node, is upper-bounded by an $O(\sqrt{n(\mathcal{N}_j)})$ term, where $n(\mathcal{N}_j)$ denotes the number of decisions made at node $\mathcal{N}_j$. Integrating the decisions made at all the nodes $\mathcal{N}_j$, $j > \kappa$, through Jensen's inequality, as discussed in Section 4.1, noticing that these nodes correspond to states in a refinement of a k-th order Markov machine, and observing that there are at most $c(x^n)$ nodes in the tree, we conclude that

$$\frac{1}{n}\left[\bar{\mathcal{L}}_{\mathrm{LZ}}^{\boldsymbol{w}}(x^n) - \bar{\mu}_{\mathcal{M}_k}^{\boldsymbol{w}}(x^n)\right] = O\left(\sqrt{\frac{c(x^n)}{n}}\right) + O\left(\frac{c(x^n)}{n}\right).$$

The theorem follows from $c(x^n) = O(n/(\log n))$ (see [10]). □

Theorems 3 and 5 imply the following Corollary.

Corollary 1 *For any infinite sequence x^∞ and any weight vector $\boldsymbol{w}$, we have*

$$\limsup_{n\to\infty} \frac{1}{n} \bar{\mathcal{L}}_{\mathrm{LZ}}^{\boldsymbol{w}}(x^n) \leq \bar{\mu}^{\boldsymbol{w}}(x^\infty).$$

Remarks.

a) Since the decisions at each node do not require sub-sampling of the corresponding subsequence of observations, as shown in Section 4.1, the upper bound on the corresponding regret is smaller than with the alphabet extension suggested by Theorem 2. While this scheme is not shown to yield lower regret than the strategy based on alphabet extension and sub-sampling (especially since the asymptotic behavior is dominated by the growth of the LZ tree), it appears to perform better in practice.

b) In terms of complexity, the main advantage of this scheme over alphabet extension appears to be in memory usage. For efficient traversal of the LZ trees over the extended alphabet, each set of branches stemming from a given node can be implemented with a sub-tree of depth τ over the original alphabet (otherwise, identification of each τ-tuple would require moving back and forth in the sequence of observations). For each sample, a pointer is advanced in each of the τ parallel trees, and the values $L(\boldsymbol{Y}, b)$ are updated for each possible action b, accumulating the contribution of each component of $\boldsymbol{Y}$. When all the components of a vector $\boldsymbol{Y}$ have been observed, a decision is made at the node attained in the LZ tree corresponding to that phase, and the counts in that node are updated. The number of operations to complete this process is roughly equivalent to the τ updates into the past required by the proposed scheme. However, the size of *each* of the τ LZ trees is roughly equivalent to that of the single tree required by the proposed scheme. This claim follows from the fact that the number of phrases behaves as $n(i)/\log n(i)$, where $n(i) \approx n/\tau$ is the length of the sub-sequence corresponding to phase i, and each branch over the extended alphabet corresponds, as discussed above, to τ branches over the original alphabet.

5 Application: adaptive prefetching

To conclude this paper, we show that the delayed prediction scenario is encountered when *adaptive prefetching* strategies for computer memory architectures are formalized as a sequential decision problem. In this application, the goal is to prefetch an address from main memory into a small, faster memory ("cache") ahead of time, in order to prevent stall time by the central processing unit (CPU) when accessing this address. Notice that the following brief discussion is intended only as a motivation, thus ignoring the intricacies of various prefetching system architectures proposed in the literature.

In a simplified memory architecture, if an address requested by the CPU is neither stored in fast memory (cache miss) nor on the bus (on its way to satisfy a previous request), a memory transaction takes place. The request is placed on the bus, and the data is returned to the CPU after a *memory latency* time T_{lat}, which is the key to view this application in terms of *delayed*

prediction. The *bus residency* (i.e., the time during which the request is on the bus) is usually negligible with respect to the memory latency, so that multiple requests may co-exist. In a prefetching architecture, a *prefetcher* recognizes patterns in the sequence of references $\{x_t\}$ and speculatively requests addresses that are likely to be accessed in the future. We will assume here that at each time index t a decision to prefetch only one address is made. Therefore, we can view the prefetched address as an action b_t, and the observation space A coincides with the action space B.[6] Upon receipt, the requested data is inserted in the cache and, in principle, old addresses are replaced. For the sake of simplicity, however, we will ignore cache replacement policies by assuming a large enough cache.

If prefetched data is referenced by the CPU, the CPU stall time caused by accessing main memory is totally or partially avoided. Partial savings occur when the reference takes place less than T_{lat} time units after the referenced data was requested by the prefetcher, so that the data is still not available. Clearly, this is a delayed prediction scenario in which the prefetcher needs to account for T_{lat}, as well as for observed times between cache misses, in order to determine how many steps in advance the prediction should be issued. The loss in this sequential decision problem is given by the total CPU stall time. Thus, in principle, the instantaneous loss (CPU stall time) incurred as a result of referencing address x_t, depends not only on the last prefetching decision b_t, but on the entire sequence of actions $b_1 b_2 \cdots b_t$. Alternatively, a more tractable loss function of the form (2) for the Hamming loss (38) results from considering an accumulation of *opportunity cost* losses. Specifically, in this formulation, $w_d \ell(b, a)$ is an *architecture-dependent* measure of the stall time that could have been saved by prefetching a instead of b at time index t, given that a occurred at time $t + d$, $0 \leq d < \tau$. The weights reflect the relation between T_{lat} and the interval lengths between misses (other system parameters may also be incorporated), and can therefore vary arbitrarily, depending not only on d but also on t. This dependency is given by side information independent of the actions $\{b_t\}$, and is revealed to the decision maker at times $t + d$, *after* the action.

[6] In a practical system, the sequence $\{x_t\}$ will typically be given by the sequence of cache misses. Moreover, locality can be exploited by defining the problem over address differences, similar to the use of prediction in the compression of smooth data (thus working "in the DPCM domain"). This technique effects a reduction of the alphabets A and B, allowing to overcome the high learning costs associated with large alphabets. Here, this differentiation process is disregarded.

The sequential decision problem has customarily been treated in the context of *repeated play* (see, e.g., [1]), where the decision maker wishes to approximate a Bayes envelope by playing the same game over time, with a fixed loss function. The fixed loss assumption is also made in the setting of learning with expert advice, but it is interesting to notice that it is not actually required in that setting. The decision strategy resulting from the exponential weighting algorithm depends only on the loss accumulated by each expert over the past, and only assumes that this loss is available at the time the decision is made. In particular, it is irrelevant whether this loss originates from a fixed loss function or from a sequence of loss functions, as long as this sequence is the same for every expert, and it is uniformly bounded. In fact, the proof given in [19] of the convergence of the normalized loss to the normalized loss of the best expert (for finite alphabets) holds *verbatim* when the assumption of a fixed loss is removed.[7]

For a given FSM, the above generalization applies to the sub-sequence occurring at each state. In the delayed prediction case with fixed loss function but variable weights, the loss achieved by the best FSM reference strategy is no longer given by (26). Instead, for a sequence of weight vectors $\{\boldsymbol{w}\}_t$, where each vector is denoted $(w_{t,0}, w_{t,1}, \cdots, w_{t,\tau-1})$, we have

$$\bar{\mu}_{f,s_1}^{\{\boldsymbol{w}\}}(x^n) = \sum_{s\in S} p_{x^{n-\tau+1}}(s) \min_{b\in B} \left\{ \sum_{d=0}^{\tau-1} \sum_{t\,:\,s_t=s} w_{t,d}\ell(b, x_{t+d}) \right\} . \tag{42}$$

The proof of Theorem 5 carries over to this case, provided that the weights remain bounded. Thus, even under these variable conditions, the on-line scheme can still compete successfully against Markov strategies. On the other hand, however, Theorem 3 *does not* carry over to this case, so that the LZ-based scheme may fail against more general FSM strategies. We conclude this section with an example showing that for a particular sequence of weight vectors, there indeed exist FSM's that outperform any Markov machine.

Example. Let $\tau = 2$, and consider two weight vectors $\boldsymbol{w}^{(1)} = (0, 1)$ and $\boldsymbol{w}^{(2)} = (1, 0)$, under Hamming loss. While $\boldsymbol{w}^{(1)}$ corresponds to a unit delay in the prediction, $\boldsymbol{w}^{(2)}$ corresponds to non-delayed prediction. Given a (large) integer N, assume that $\boldsymbol{w_t} = \boldsymbol{w}^{(1)}$ when $\lfloor t/N \rfloor$ is even, and $\boldsymbol{w_t} = \boldsymbol{w}^{(2)}$ otherwise (i.e., the vector remains constant for N time instants, and alternates between $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$).

[7] While this observation is not true for the algorithm proposed in [1], it was shown in [20, Lemma 1] that a simple modification of this algorithm can be used with varying losses.

Now, consider the binary sequence $x^n = 010101\cdots$. Clearly, for this sequence, a Markov predictor of any given order k will alternate between two states, denoted $s^{(1)}$ and $s^{(2)}$, for all $t > k$, *regardless of* k. If the predictions $g(s^{(1)})$ and $g(s^{(2)})$ for these states differ, the Markov strategy will also alternate its predictions for $t > k$. In this situation, the loss will always be either 0 under $\boldsymbol{w}^{(1)}$ and 1 under $\boldsymbol{w}^{(2)}$, or vice versa. If, instead, the predictions $g(s^{(1)})$ and $g(s^{(2)})$ coincide, a constant predictor will incur a loss every other time for both $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$, except for the transitions between the loss phases (every N time instants). Since N is large, in both cases the normalized loss approaches 0.5.

In contrast, an FSM strategy can track the variation of the loss function and adjust the phase of its alternating predictions as the loss function changes, to achieve virtually 0 loss (again, with the negligible exception of the transitions between loss functions).

The above example can be modified to show that for variable loss functions in the non-delayed case, FSM strategies can outperform any Markov strategy. Let $\ell_1(\cdot,\cdot)$ denote the Hamming loss function for the binary case, let $\ell_2(\cdot,\cdot) = 1 - \ell_1(\cdot,\cdot)$, and assume that the two loss functions alternate. Clearly, for the all-zero sequence, any Markov strategy will remain in the same state for $t > k$, and will therefore incur a loss every other symbol. In contrast, an FSM strategy can adapt to the varying loss function to achieve 0 loss. It should be noticed, however, that in many cases a variable loss function can be viewed as a fixed one by considering the loss as part of the observation. In the above example, given the observation and the corresponding loss value, the decision maker can infer which loss function was used. Letting the observation space be given by $\{0,1\} \times \{\ell_1, \ell_2\}$, while the action space remains $\{0,1\}$, the variable loss function clearly corresponds to a fixed one. In order to attain the asymptotic equivalence between Markov and general FSM's, the machines must be driven by the compound observations. However, in many practical applications, the corresponding extensions of the proposed on-line schemes would be prohibitively complex.

A Appendix: Delayed-mode performance of binary predictors

In this appendix we investigate the performance of the minimax binary predictor of [3], and the binary predictor resulting from exponential weighting [6], when applied with a delay d.

Minimax predictor. We first show that for any sequence x^n, the regret $R_{\mathrm{MM}}^{(d)}(x^n)$ of the minimax predictor, when applied to a delayed sequence, satisfies the upper bound

$$R_{\mathrm{MM}}^{(d)}(x^n) \leq (1+2d)\sqrt{\frac{n}{2\pi}} + o(\sqrt{n}) \tag{A.1}$$

where d is the delay. We then demonstrate a sequence that attains this upper bound asymptotically. Let $p_{t+1}^{(\mathrm{MM})}(x_{t+1}|x^{t-d}, n)$ denote the probability assigned to x_{t+1} under this scheme, with horizon n, where for $0 \leq t \leq d$, x^{t-d} is the null sequence λ, for which it is assumed that $p_{t+1}^{(\mathrm{MM})}(x_{t+1}|\lambda, n) = \frac{1}{2}$. The corresponding loss incurred on x^n is

$$\bar{\mathcal{L}}_{\mathrm{MM}}(x^n) = \sum_{t=0}^{n-1} [1 - p_{t+1}^{(\mathrm{MM})}(x_{t+1}|x^{t-d}, n)] \,. \tag{A.2}$$

Recall that, for non-delayed prediction of x_{n-r} after observing x^{n-r-1}, the scheme of [3] can be interpreted as drawing $x_{n-r+1}, \cdots, x_n$ at random, and choosing x_{n-r} as the most frequent symbol in the resulting sequence $x_1 x_2 \cdots x_{n-r-1} x_{n-r+1} \cdots x_n$ of length $n-1$ (with a coin flip deciding ties for odd n). With $n_h \stackrel{\Delta}{=} \lceil n/2 \rceil$, it is easy to see that regardless of the parity of n, the probability $q_{r,n}(1)$ assigned to 1 by that scheme takes the form

$$q_{r,n}(1) = 2^{-(2n_h - n + r)} \sum_{i=n_h - n_1(x^{n-r-1})}^{2n_h - n + r} \binom{2n_h - n + r}{i} \tag{A.3}$$

where we recall that $n_a(x^t)$ denotes the number of occurrences of $a \in \{0,1\}$ in the sequence x^t. Clearly, $p_{t+1}^{(\mathrm{MM})}(1|x^{t-d}, n) = q_{n+d-t-1,n}(1)$.

Now, consider the (delayed) probability assignment for horizon $2n_h + d$ at time $t+d+1$, after observing a sequence of length t composed of x^{t-d} followed by d copies of x_{t+1}. Using (A.3) and denoting $d_h \stackrel{\Delta}{=} \lceil d/2 \rceil$, it can be verified (e.g., by distinguishing between the even and odd d cases, and using the addition formula for combinatorial coefficients for the latter case) that this assignment takes the form

$$\begin{aligned} & p_{t+d+1}^{(\mathrm{MM})}(x_{t+1}|x^{t-d}x_{t+1}\cdots x_{t+1}, 2n_h + d) = 2^{-(2n_h - t + d - 1)} \cdot \\ & \left[\sum_{i=n_h + d_h - d - n_{x_{t+1}}(x^{t-d})}^{2n_h - t + d - 1} \binom{2n_h - t + d - 1}{i} + (d_h - \frac{d}{2}) \binom{2n_h - t + d - 1}{n_h - d_h - n_{x_{t+1}}(x^{t-d})} \right] \\ \leq \; & 2^{-(2n_h - t + d - 1)} \cdot \left[\sum_{i=n_h - n_{x_{t+1}}(x^{t-d})}^{2n_h - t + d - 1} \binom{2n_h - t + d - 1}{i} + \frac{d}{2} \binom{2n_h - t + d - 1}{\lfloor n_h - \frac{t}{2} + \frac{d}{2} - \frac{1}{2} \rfloor} \right] \end{aligned} \tag{A.4}$$

where the inequality follows from the fact that for any pair of integers m and i, $0 \le i \le m$, we have $\binom{m}{i} \le \binom{m}{\lfloor m/2 \rfloor}$. Thus, using (A.3) with $r = n+d-t-1$, we obtain

$$\begin{aligned} p^{(\mathrm{MM})}_{t+d+1}(x_{t+1}|x^{t-d}x_{t+1}\cdots x_{t+1}, 2n_h+d) &\le p^{(\mathrm{MM})}_{t+1}(x_{t+1}|x^{t-d}, n) \\ &+ \frac{d}{2}\, 2^{-(2n_h-t+d-1)} \binom{2n_h-t+d-1}{\lfloor n_h - \frac{t}{2} + \frac{d}{2} - \frac{1}{2} \rfloor} \\ &\le p^{(\mathrm{MM})}_{t+1}(x_{t+1}|x^{t-d}, n) + \frac{d}{2}\, 2^{-(2n_h-t+d-1)} \binom{n-t+d}{\lfloor \frac{n-t+d}{2} \rfloor} \\ &\le p^{(\mathrm{MM})}_{t+1}(x_{t+1}|x^{t-d}, n) + \frac{d}{2}\sqrt{\frac{2}{\pi(n-t+d)}} \end{aligned} \tag{A.5}$$

where the last inequality follows from Stirling's formula. Therefore, an upper bound on the loss in (A.2) is given by

$$\bar{\mathcal{L}}_{\mathrm{MM}}(x^n) \le \sum_{t=0}^{n-1} \left[1 - p^{(\mathrm{MM})}_{t+d+1}(x_{t+1}|x^{t-d}x_{t+1}\cdots x_{t+1}, 2n_h+d) + \frac{d}{2}\sqrt{\frac{2}{\pi(n-t+d)}} \right]. \tag{A.6}$$

Now, a key observation in the derivation of (A.1) is that

$$p^{(\mathrm{MM})}_{t+d+1}(x_{t+1}|x^{t-d}x_{t+1}\cdots x_{t+1}, 2n_h+d) \ge p^{(\mathrm{MM})}_{t+d+1}(x_{t+1}|x^t, 2n_h+d)$$

since additional occurrences of x_{t+1} cannot decrease the probability assigned to it. Hence, (A.6) takes the form

$$\begin{aligned} \bar{\mathcal{L}}_{\mathrm{MM}}(x^n) &\le \sum_{t=0}^{n-1} [1 - p^{(\mathrm{MM})}_{t+d+1}(x_{t+1}|x^t, 2n_h+d)] + \frac{d}{\sqrt{2\pi}} \sum_{j=d+1}^{n+d} \frac{1}{\sqrt{j}} \\ &\le \bar{\mathcal{L}}_{0,2n_h+d}(x^n) + \frac{d}{\sqrt{2\pi}} \int_0^n \frac{dx}{\sqrt{x}} \end{aligned} \tag{A.7}$$

where $\bar{\mathcal{L}}_{0,2n_h+d}(x^n)$ denotes the loss incurred on x^n by the minimax scheme *without* delay, but with a horizon $2n_h+d$. Clearly, $\bar{\mathcal{L}}_{0,2n_h+d}(x^n) \le \bar{\mathcal{L}}_{0,2n_h+d}(x^n a^{d+2n_h-n})$, where a denotes the most frequent symbol in x^n and a^m denotes m copies of a. Recalling that $B(x^n) = \min(n_0(x^n), n_1(x^n))$, for any non-negative integer m we have $B(x^n) = B(x^n a^m)$. Thus, by Lemma 1,

$$\bar{\mathcal{L}}_{0,2n_h+d}(x^n) \le B(x^n) + R_0(d+2n_h)$$

and since the (non-delayed) minimax regret for horizon m, $R_0(m)$, is non-decreasing with m, (A.7) implies

$$\bar{\mathcal{L}}_{\mathrm{MM}}(x^n) \le B(x^n) + R_0(n+d+1) + d\sqrt{\frac{2n}{\pi}}.$$

The claimed bound (A.1) follows from the asymptotic expansion of the explicit expression (8) for $R_0(n)$.

Next, we show that for any sufficiently large n, the upper bound (A.1) is attained by the sequence $y^n = (1^h 0^h)^m$, composed of m copies of a block formed by h ones followed by h zeroes, where h and m satisfy $n = 2hm$, $h \gg d$, and will otherwise be specified later. To this end, we will compare the performances of the predictor used in delayed and non-delayed mode. After observing y^{t-d}, the predicted value is compared to y_{t+1} in the delayed case, and to y_{t-d+1} in the non-delayed case. The key observation is that $y_{t+1} = y_{t-d+1}$ for all $t \geq d$, except for those values of t that take the form $t = kh+i$, for any positive integer k and any integer i in the range $0 \leq i < d$. Now, for odd k, the delayed predictor observes $y_{t+1} = 0$, whereas the non-delayed one would have observed the symbol $y_{t-d+1} = 1$, which was assigned a larger probability, as it was the most frequently observed symbol at that point. In contrast, for even k, the delayed predictor yields a smaller loss, as it observes the (most frequent) symbol $y_{t+1} = 1$, whereas the non-delayed one would have observed $y_{t-d+1} = 0$. It is easy to see that the difference $\Delta\bar{\mathcal{L}}(y^n)$ between the losses incurred by the delayed and the non-delayed predictor satisfies

$$\Delta\bar{\mathcal{L}}(y^n) > \sum_{j=0}^{m-1}\sum_{i=0}^{d-1}[q_{r(i,j),n}(1) - q_{r(i,j),n}(0)] - \sum_{j=1}^{m-1}\sum_{i=0}^{d-1}[q_{r(i,j)-h,n}(1) - q_{r(i,j)-h,n}(0)] \qquad \text{(A.8)}$$

where $r(i,j) \stackrel{\Delta}{=} (2j+1)h + d - i - 1$. Notice that the discrepancy between the left-hand and right-hand sides of (A.8) is due to the fact that the loss difference for the first $d+1$ symbols in the sequence, which is positive, has not been accounted for. Since n is even, by (A.3) we have

$$\begin{aligned} q_{r(i,j),n}(1) - q_{r(i,j),n}(0) &= 2^{-r(i,j)} \sum_{s=jh+d-i}^{(j+1)h-1} \binom{r(i,j)}{s} \\ &> (h-d+i)2^{-r(i,j)} \binom{r(i,j)}{jh+d-i}. \end{aligned} \qquad \text{(A.9)}$$

For positive integers u and v, it can readily be verified that

$$2^{-u}\binom{u}{v} < 2^{-(u+1)}\binom{u+1}{v+1}$$

provided that $u \geq 2v+1$. Thus, since $h \gg d$, (A.9) implies

$$q_{r(i,j),n}(1) - q_{r(i,j),n}(0) > (h-d)2^{-(2j+1)h}\binom{(2j+1)h}{jh}.$$

Similarly, it can be shown that

$$q_{r(i,j)-a,n}(1) - q_{r(i,j)-a,n}(0) < d2^{-2jh}\binom{2jh}{jh}.$$

Hence, (A.8) takes the form

$$\Delta\bar{\mathcal{L}}(y^n) > d(h-d)\sum_{j=0}^{m-1} 2^{-(2j+1)h}\binom{(2j+1)h}{jh} - d^2\sum_{j=1}^{m-1} 2^{-2jh}\binom{2jh}{jh}. \tag{A.10}$$

Now, for positive integers u and v, the Gaussian approximation of the binomial coefficients states (see, e.g., [21, Chapter 9])

$$\binom{2u}{u-v} = \frac{2^{2u}}{\sqrt{\pi u}}e^{-v^2/u}[1+O(1/\sqrt{u})]$$

for all $v < \sqrt{u}$. Thus, for all $j > h/4$, we have

$$2^{-(2j+1)h}\binom{(2j+1)h}{jh} = \frac{e^{-h/(4j)}}{\sqrt{\pi jh}}[1+O(1/\sqrt{jh})]. \tag{A.11}$$

Further assuming, e.g., $j > h^2/4$, it is easy to see that (A.11) implies

$$2^{-(2j+1)h}\binom{(2j+1)h}{jh} > \frac{1-O(1/h)}{\sqrt{\pi jh}}. \tag{A.12}$$

In addition, the negative terms in the right-hand side of (A.10) can be upper-bounded using Stirling's approximation as in (A.5), which together with (A.12) implies

$$\begin{aligned}\Delta\bar{\mathcal{L}}(y^n) &> d(h-d)\frac{1-O(1/h)}{\sqrt{\pi h}}\sum_{j=h^2/4}^{m-1}\frac{1}{\sqrt{j}} - \frac{d^2}{2\sqrt{\pi h}}\sum_{j=1}^{m-1}\frac{1}{\sqrt{j}} \\ &\geq \frac{2d(h-d)}{\sqrt{\pi h}}[1-O(1/h)][\sqrt{m}-\frac{h}{2}] - \frac{d^2}{\sqrt{\pi h}}\sqrt{m}. \end{aligned}\tag{A.13}$$

Choosing h to be a function of n such that $h \to \infty$ when $n \to \infty$, and $h = o(\sqrt{m})$ (so that $h = o(n^{1/3})$), it follows that

$$\Delta\bar{\mathcal{L}}(y^n) > d\sqrt{\frac{2n}{\pi}} - o(\sqrt{n}).$$

Since $R_{\mathrm{MM}}^{(d)}(y^n) = R_0(n) + \Delta\bar{\mathcal{L}}(y^n)$, we obtain

$$R_{\mathrm{MM}}^{(d)}(y^n) > (1+2d)\sqrt{\frac{n}{2\pi}} - o(\sqrt{n})$$

as claimed.

Exponential weighting. Next, we consider the exponential weighting algorithm in the context of constant experts, and we start by showing that for any sequence x^n, the regret $R_{\mathrm{EW}}^{(d)}(x^n)$ of the corresponding binary predictor applied to a delayed sequence satisfies the upper bound

$$R_{\mathrm{EW}}^{(d)}(x^n) \leq \sqrt{\frac{(1+2d)n\ln 2}{2}} + O(1) \tag{A.14}$$

for a suitable choice of the weighting parameter η. We then demonstrate a sequence that attains this upper bound asymptotically.

Recall that for given η, the probability assigned to 1 by the exponential weighting predictor (for constant experts) after observing x^t is given by

$$p^{(\mathrm{EW})}(1|x^t) = \frac{1}{1+e^{-\eta\delta(1|x^t)}}$$

where, for $a \in \{0,1\}$, $\delta(a|x^t) \stackrel{\Delta}{=} n_a(x^t) - n_{1-a}(x^t)$. As in the case of the minimax algorithm, let $\Delta\bar{\mathcal{L}}(x^n)$ denote the difference between the losses incurred by the delayed and the non-delayed predictor on x^n. We have

$$\Delta\bar{\mathcal{L}}(x^n) = \sum_{t=0}^{n-1}[p^{(\mathrm{EW})}(x_{t+1}|x^t) - p^{(\mathrm{EW})}(x_{t+1}|x^{t-d})]\,.$$

Clearly, if we append d copies of x_{t+1} to x^{t-d}, we have

$$p^{(\mathrm{EW})}(x_{t+1}|x^{t-d}x_{t+1}\cdots x_{t+1}) \geq p^{(\mathrm{EW})}(x_{t+1}|x^t)\,.$$

Therefore,

$$\begin{aligned}\Delta\bar{\mathcal{L}}(x^n) &\leq \sum_{t=0}^{n-1}[p^{(\mathrm{EW})}(x_{t+1}|x^{t-d}x_{t+1}\cdots x_{t+1}) - p^{(\mathrm{EW})}(x_{t+1}|x^{t-d})] \\ &= \sum_{t=0}^{n-1}\left[\frac{1}{1+e^{-\eta(\delta(x_{t+1}|x^{t-d})+d)}} - \frac{1}{1+e^{-\eta\delta(x_{t+1}|x^{t-d})}}\right] \stackrel{\Delta}{=} \sum_{t=0}^{n-1}\Delta_t\,. \end{aligned} \tag{A.15}$$

It is easy to verify that Δ_t is maximum for $\delta(x_{t+1}|x^{t-d}) = -d/2$. Consequently,

$$\Delta\bar{\mathcal{L}}(x^n) \leq \frac{n(e^{d\eta/2}-1)}{2}\,. \tag{A.16}$$

Now, for sufficiently small η and any constant $K > \frac{1}{2}$, we have

$$e^{d\eta/2} \leq 1 + \frac{\eta d}{2} + \frac{K\eta^2 d^2}{4}\,.$$

(E.g., for $K = e - 2$ it suffices to assume $\eta \leq 2/d$.) Thus, for η in that range, (A.16) further implies

$$\Delta\bar{\mathcal{L}}(x^n) \leq n\left(\frac{\eta d}{4} + \frac{K\eta^2 d^2}{8}\right). \tag{A.17}$$

Since the regret incurred by a (non-delayed) application of the exponential weighting predictor on a sequence of length n is upper-bounded by $(n\eta/8) + (\ln 2)/\eta$ (see, e.g., [19]), for delayed prediction (A.17) yields the bound

$$R_{\mathrm{EW}}^{(d)}(x^n) \leq \frac{\eta n}{8} + \frac{\ln 2}{\eta} + \frac{\eta d n}{4} + \frac{K\eta^2 d^2 n}{8}.$$

Finally, (A.14) follows from choosing, for a given horizon,

$$\eta = \sqrt{\frac{8\ln 2}{(2d+1)n}}$$

which, for sufficiently large n, indeed satisfies the above condition on the range of η. Notice that if d is not known to the predictor, which chooses to use the same value of η that is optimum for the non-delayed case (namely, $\eta = \sqrt{8(\ln 2)/n}$), the asymptotic upper bound on the regret is $(d+1)\sqrt{n(\ln 2)/2}$. Thus, the performance is still better than with the minimax scheme of [3] for any $d > 0$.

Finally, we show that for *any* value of η and any sufficiently large n, there exists a sequence y^n that attains the upper bound (A.14). The sequence takes the form $y^n = (1^h 0^h)^m 0^z$, where the integers h, m, and z, depend on η, satisfy $n = 2hm + z$, and will otherwise be specified later. Notice that this sequence differs from the one used for the minimax algorithm in that it contains a tail composed of z zeroes, following the m blocks of the form $1^h 0^h$. The reason for this tail is that, otherwise, a weighting parameter value $\eta = 0$ would trivially suffice to approach the Bayes response with zero regret.

Studying the evolution of the value of $\delta(1|y^t)$ across the sequence, and assuming $h \gg d$, it can readily be verified that the loss $\bar{\mathcal{L}}_{\mathrm{EW}}(y^n)$ incurred on y^n by the delayed predictor under consideration satisfies

$$\begin{aligned}\bar{\mathcal{L}}_{\mathrm{EW}}(y^n) &= m\left[\sum_{i=1}^{d}\frac{1}{1+e^{\eta i}} + \sum_{i=0}^{h-d-1}\frac{1}{1+e^{\eta i}} + \sum_{i=d+1}^{h}\frac{1}{1+e^{-\eta i}} + \sum_{i=h-1}^{h-d}\frac{1}{1+e^{-\eta i}}\right] \\ &+ \left[\frac{d}{2} - \sum_{i=1}^{d}\frac{1}{1+e^{\eta i}}\right] + \sum_{i=-d}^{z-d-1}\frac{1}{1+e^{\eta i}}.\end{aligned} \tag{A.18}$$

In (A.18), the term that appears m times corresponds to a typical block of the form 1^h0^h, the difference that follows is a correction term for the first d symbols in the sequence, for which the behavior differs from the typical block, and the last summation corresponds to the all-zero tail. After some algebraic manipulation, (A.18) takes the form

$$\begin{aligned} R_{\mathrm{EW}}^{(d)}(y^n) &= \bar{\mathcal{L}}_{\mathrm{EW}}(y^n) - mh = m\sum_{i=1}^{d}\left[\frac{1-e^{\eta i}}{1+e^{\eta i}} + \frac{1-e^{-\eta(h-i)}}{1+e^{-\eta(h-i)}}\right] + \frac{m(1-e^{-\eta h})}{2(1+e^{-\eta h})} \\ &+ \left[\frac{d}{2} - \sum_{i=1}^{d}\frac{1}{1+e^{\eta i}}\right] + \sum_{i=-d}^{z-d-1}\frac{1}{1+e^{\eta i}}\,. \end{aligned} \tag{A.19}$$

Notice that we can assume η to be a vanishing function of n, for otherwise, taking $h = d+1$ and $z = 0$, it is easy to see that the normalized regret does not even vanish. Moreover, given η, we will choose h such that when $n \to \infty$, $h \to \infty$ but $h\eta$ vanishes. Thus, for sufficiently large n, we can use the Taylor expansion of the function $(1-e^x)/(1+e^x)$ to further lower-bound the right-hand side of (A.19), obtaining

$$\begin{aligned} R_{\mathrm{EW}}^{(d)}(y^n) &> m\sum_{i=1}^{d}\left[-\frac{\eta i}{2} + \frac{\eta(h-i)}{2} - \frac{\eta^3(h-i)^3}{24}\right] + \frac{m\eta h}{4} - \frac{m\eta^3 h^3}{48} + \int_{-d}^{z-d-1}\frac{dx}{1+e^{\eta x}} \\ &= \frac{\eta(n-z)(2d+1)}{8} + \frac{1}{\eta}\ln(1+e^{\eta d}) - \frac{1}{\eta}\ln(1+e^{-\eta(h-d)}) - mo(h\eta)\,. \end{aligned} \tag{A.20}$$

Now, choose $z = \eta^{-1}\ln\eta^{-1}$. After straightforward manipulations, (A.20) yields

$$R_{\mathrm{EW}}^{(d)}(y^n) > \frac{\eta n(2d+1)}{8} + \frac{\ln 2}{\eta} - \eta o(n) - O\left(\ln\frac{1}{\eta}\right)\,. \tag{A.21}$$

The result follows from minimizing the right-hand side of (A.21) with respect to η. Again, if the given η is the optimal value for non-delayed prediction, the attained asymptotic bound on the regret is $(d+1)\sqrt{n(\ln 2)/2}$.

B Appendix: Proof of Lemma 2

Let $s(j)$ denote the j-tuple corresponding to a refined state $s \in S$. We have

$$\begin{aligned} \bar{\mu}_{\mathcal{M}_j}^{\boldsymbol{w}}(x^n) - \bar{\mu}_f^{\boldsymbol{w}}(x^n) &= \sum_{s\in S} p_{x^{n-\tau+1}}(s)\left[\min_{b\in B}\left\{\sum_{d=0}^{\tau-1} w_d \sum_{u\in A^d}\sum_{a\in A} p_{x^n(\tau,d)}(ua|s(j))\ell(b,a)\right\}\right] \\ &- \sum_{s\in S} p_{x^{n-\tau+1}}(s)\left[\min_{b\in B}\left\{\sum_{d=0}^{\tau-1} w_d \sum_{u\in A^d}\sum_{a\in A} p_{x^n(\tau,d)}(ua|s)\ell(b,a)\right\}\right] \end{aligned} \tag{B.22}$$

where $n(\tau,d) \stackrel{\Delta}{=} n-\tau+d+1$. Let $b(s)$ denote the minimizing action for $s \in S$ in the *second* minimum in the right-hand side of (B.22). It follows that

$$\begin{aligned}
\bar{\mu}^{\boldsymbol{w}}_{\mathcal{M}_j}(x^n) - \bar{\mu}^{\boldsymbol{w}}_f(x^n) &\leq \sum_{s\in S} p_{x^{n-\tau+1}}(s) \left[\sum_{d=0}^{\tau-1} w_d \sum_{u\in A^d} \sum_{a\in A} |p_{x^{n(\tau,d)}}(ua|s(j)) - p_{x^{n(\tau,d)}}(ua|s)| \ell(b(s),a) \right] \\
&\leq \ell_{\max} \sum_{d=0}^{\tau-1} w_d \sum_{s\in S} p_{x^{n-\tau+1}}(s) \sum_{v\in A^{d+1}} |p_{x^{n(\tau,d)}}(v|s(j)) - p_{x^{n(\tau,d)}}(v|s)| \\
&= \ell_{\max} \sum_{d=0}^{\tau-1} w_d \sum_{s\in S} p_{x^{n-\tau+1}}(s) \sum_{v\in A^{d+1}} \left| \sum_{z\in A^{\tau-1-d}} [p_{x^n}(vz|s(j)) - p_{x^n}(vz|s)] \right| \\
&\leq \ell_{\max} W \sum_{s\in S} p_{x^{n-\tau+1}}(s) \sum_{y\in A^{\tau}} |p_{x^n}(y|s(j)) - p_{x^n}(y|s)| \,. \qquad \text{(B.23)}
\end{aligned}$$

The lemma follows from Pinsker's inequality [18, Chapter 3, Problem 17].

Acknowledgments

Many thanks to Gadiel Seroussi and Neri Merhav for useful discussions, to Tomas Rokicki for his insight in the prefetching application, and to the anonymous reviewers for their helpful comments.

References

[1] J. F. Hannan, "Approximation to Bayes risk in repeated plays," in *Contributions to the Theory of Games, Volume III, Ann. Math. Studies*, vol. 3, pp. 97–139, Princeton, NJ, 1957.

[2] D. Blackwell, "An analog for the minimax theorem for vector payoffs," *Pacific J. Math.*, vol. 6, pp. 1–8, 1956.

[3] T. M. Cover, "Behavior of sequential predictors of binary sequences," in *Proc. 4th Prague Conf. Inform. Theory, Statistical Decision Functions, Random Processes*, (Prague), pp. 263–272, Publishing House of the Czechoslovak Academy of Sciences, 1967.

[4] M. Feder, N. Merhav, and M. Gutman, "Universal prediction of individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 1258–1270, July 1992.

[5] V. G. Vovk, "Aggregating strategies," in *Proc. of the 3rd Annual Workshop on Computational Learning Theory*, (San Mateo, California), pp. 372–383, 1990.

[6] N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, D. Haussler, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *Journal of the ACM*, vol. 44, no. 3, pp. 427–485, 1997.

[7] Y. M. Shtarkov, "Universal sequential coding of single messages," *Problems of Inform. Trans.*, vol. 23, pp. 175–186, July 1987.

[8] T. H. Chung, *Minimax Learning in Iterated Games via Distributional Majorization*. PhD thesis, Department of Electrical Engineering, Stanford University, 1994.

[9] N. Merhav and M. Feder, "Universal schemes for sequential decision from individual data sequences," *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 1280–1292, July 1993.

[10] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 530–536, Sept. 1978.

[11] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, pp. 212–261, 1994.

[12] D. Haussler, J. Kivinen, and M. K. Warmuth, "Sequential prediction of individual sequences under general loss functions," *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 1906–1925, Sept. 1998.

[13] N. Merhav, E. Ordentlich, G. Seroussi, and M. J. Weinberger, "On sequential strategies for loss functions with memory," Jan. 2001. Submitted to IEEE Trans. Inform. Theory.

[14] J. Rissanen, "A universal data compression system," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 656–664, Sept. 1983.

[15] J. S. Vitter and P. Krishnan, "Optimal prefetching via data compression," in *Proc. 1991 Conf. Foundation of Computer Sci. (FOCS)*, pp. 121–130, 1991.

[16] R. E. Krichevskii and V. K. Trofimov, "The performance of universal encoding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 199–207, Mar. 1981.

[17] M. J. Weinberger, N. Merhav, and M. Feder, "Optimal sequential probability assignment for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 384–396, Mar. 1994.

[18] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. New York: Academic, 1981.

[19] N. Cesa-Bianchi and G. Lugosi, "On prediction of individual sequences," *Annals of Statistics*, vol. 27, no. 6, 1999.

[20] M. J. Weinberger and E. Ordentlich, "On-line decision making for a class of loss functions via Lempel-Ziv parsing," in *Proc. 2000 Data Compression Conference*, (Snowbird, Utah, USA), pp. 163–172, Mar. 2000.

[21] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Reading, MA: Addison-Wesley Publishing Company, 1989.