Finite-Memory Universal Prediction of Individual Sequences

Eado Meron and Meir Feder, *Fellow, IEEE*

***Abstract*—The problem of predicting the next outcome of an individual binary sequence under the constraint that the universal predictor has a finite memory, is explored. In this analysis, the finite-memory universal predictors are either deterministic or random time-invariant finite-state (FS) machines with K states (K-state machines). The paper provides bounds on the asymptotic achievable regret of these constrained universal predictors as a function of K, the number of their states, for long enough sequences. The specific results are as follows. When the universal predictors are deterministic machines, the comparison class consists of constant predictors, and prediction is with respect to the 0–1 loss function (Hamming distance), we get tight bounds indicating that the optimal asymptotic regret is $1/(2K)$. In that case of K-state deterministic universal predictors, the constant predictors comparison class, but prediction is with respect to the self-information (code length) and the square-error loss functions, we show an upper bound on the regret (coding redundancy) of $O(K^{-2/3})$ and a lower bound of $\Theta(K^{-4/5})$. For these loss functions, if the predictor is allowed to be a random K-state machine, i.e., a machine with random state transitions, we get a lower bound of $\Theta\left(\frac{1}{K}\right)$ on the regret, with a matching upper bound of $O\left(\frac{1}{K}\right)$ for the square-error loss, and an upper bound of $O\left(\frac{\log K}{K}\right)$[1] for the self-information loss. In addition, we provide results for all these loss functions in the case where the comparison class consists of all predictors that are order-L Markov machines.**

***Index Terms*—Exponentially decaying memory, finite-state (FS) machines, FS prediction, imaginary sliding window, saturated counter (SC), universal coding, universal prediction.**

I. Introduction

UNIVERSAL prediction and universal coding is a mature subject nowadays (see, e.g., [14] for an extensive survey). The important results are widely known, demonstrating the often surprising phenomena that it is possible to universally predict (or compress) data generated by an unknown source, or even an individual deterministic data sequence and attain optimal asymptotic performance. The universal prediction problem can be considered in both a probabilistic and a deterministic setting. In the probabilistic setting of the problem, it is assumed that the data is generated by some (unknown) probabilistic source. If the source were known, one could design an optimal (nonuniversal) predictor that would minimize the expected prediction loss for that source. Interestingly, universal prediction theory has shown that one can construct a single universal predictor that can work well for all sources, i.e., attain asymptotically the same expected average loss as the optimal predictor tuned to the source. The existence of a universal predictor requires that the unknown source belongs to a constrained enough class of sources. Also, the rate of convergence depends on richness of that class. Yet, at least for weak convergence, this class can be all finite-alphabet stationary and ergodic sources, see [10], [18].

In the deterministic setting of the universal prediction problem the data is an arbitrary individual sequence. If the data sequence is known upfront, one can choose the best predictor from some constrained class of predictors that minimizes the prediction loss for that sequence. This predictor is nonuniversal, as it is designed based on the given sequence. Interestingly, as was shown by universal prediction theory, there exists a single universal predictor whose performance for any sequence is asymptotically the same as the performance of the (nonuniversal) predictor tuned to that sequence. The existence of such universal predictor requires that the class of predictors from which this nonuniversal predictor is chosen is constrained enough. As above, the convergence rate depends on the richness of that class. Yet this class can be large, e.g., all finite-state machines, see [6].

In a further examination of the universal predictors that attain the optimal performance, it turns out that these predictors, while universal and nonanticipating, are much more complex than the predictors or the class of sources whose performance they attain. For example, optimal universal prediction and universal coding of binary sequences for memoryless sources, or predictors that compete with the class of constant predictors (there is a duality between these two problems, see [14]) must maintain the empirical count of zeros and ones observed so far in the sequence. This requires that the universal predictor will have a growing number of states (roughly n^2, where n is the data size) and all this complexity is required to compete with a constant single-state predictor!

Following this observation, a natural question arises. What is the best that can be done if the universal predictor has limited resources? For this we consider the case where the universal

Manuscript received March 31, 2003; revised March 31, 2004. The work of E. Meron is supported in part by Intel Israel, and by the "Yitzhak and Chaya Weinstein Institute for Research in Signal Processing." The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Yokohama, Japan, June/July 2003 and at the Data Compression Conference, Snowbird, UT, March 2004.

The authors are with the Department of Electrical Engineering–Systems, Tel-Aviv University, Ramat-Aviv 69978, Israel (e-mail: eado@eng.tau.ac.il; meir@eng.tau.ac.il).

Communicated by E.-h. Yang, W. Szpankowski, and J. C. Kieffer, Guest Editors.

Digital Object Identifier 10.1109/TIT.2004.830749

[1]Throughout the paper

$$f(K) = O(g(K)) \Rightarrow \limsup_{K\to\infty} \frac{f(K)}{g(K)} \leq \text{const}$$

$$f(K) = \Theta(g(K)) \Rightarrow \lim_{K\to\infty} \frac{f(K)}{g(K)} = \text{const}, \quad \log x = \log_2 x.$$

predictor has a finite number of states and is time invariant for most cases. This restriction is motivated by two main reasons: The first is a notion of "fair play." In the deterministic setting we consider the comparison classes of the constant (single-state) predictors, Markov predictors, and finite-state (FS) predictors with a limited number of states. Thus, it is only fair to use finite-memory universal predictor. In this game, the nonuniversal player has the advantage of knowing the sequence and picking the most suitable predictor for it from the reference class, while the universal player, that must pick the same predictor for all sequences has the advantage of having more states (but not infinitely more). The second reason is a practical one. Considering different loss functions, universal prediction results apply to problems such as branch prediction [5], page prefetching [22], gambling and portfolio selection [4], universal data compression [23], and so on. In some of these applications memory size and the number of computations are very costly.

In this paper, we thus consider the case where the universal predictor is constrained to be a FS machine; actually, it is either a K-state, time-invariant, deterministic finite-state (TIDFS) machine or a randomized K-state time-invariant machine. There are previous results for this problem in the probabilistic setting ([17], [7]). In this paper, however, we focus on the deterministic setting. We explore the performance of the K-state universal predictor as compared with the performance of the best predictor in the comparison class, for long enough sequences, in terms of the number of states K. We consider prediction with the 0–1 (Hamming) loss, the self-information loss (essentially the coding problem), and the square-error loss (least square prediction). The main findings are as follows.

1) For the 0–1 loss and the comparison class of constant predictors, we construct a TIDFS machine, the linear output saturated counter (LOSC), whose extra loss is $1/(2K)$. We also show a lower bound of $1/(2K)$ for the extra loss indicating that this is the optimal achievable asymptotic expected regret (AER).

2) For the self-information and square-error loss and the comparison class of constant predictors (coders or estimators), we construct a TIDFS machine whose extra loss (coding redundancy) is $O(K^{-2/3})$. We then show a lower bound for TIDFS machines of $\Theta(K^{-4/5})$. While the optimal performance for TIDFS machines is still unknown, we conjecture that $O(K^{-2/3})$ is optimal. In any case, these results show that constrained universal coding behaves differently in the deterministic and the probabilistic settings (this is not true for the unconstrained case).

3) For the problems of coding and least-square predicting using *random* K-state machines, we consider the "imaginary sliding-window" (ISW) machine, which is a known, interesting, random machine proposed, e.g., in [12], [20]. We show that the ISW machine obtains a least square regret of $O(\frac{1}{K})$ (which is optimal) and a coding redundancy which is at most $O(\frac{\log K}{K})$.

4) We provide results for all considered loss functions in the case where the comparison class consists of the order-L Markov machines, and the class of S-state machines.

In Section II, we present definition and notations including the explicit definition of FS and Markov predictors. Then, in Section III, we briefly present the previous results in the probabilistic setting that were not widely published so far. The novel results in the deterministic setting on the 0–1 loss are given in Section IV, while the results on the self-information and the square-error loss are covered in Section V. The paper is summarized, and further research is suggested, in Section VI.

II. Definitions and Notation

In this section, we set the notation and briefly describe prediction and universal prediction of binary sequences. We define the term "regret" used extensively in the paper and provide an explicit definition of FS predictors.

Throughout the paper, a predictor is a machine that receives a binary sequence $x_1, x_2, \ldots$, and at each time instant n, after having seen $x_1^{n-1} = (x_1, \ldots, x_{n-1})$ it predicts the next outcome x_n. This prediction, denoted $b_n = \hat{x}_n$, can be deterministic or random. There is a loss function $l(b_n, x_n)$ associated with b_n and the actual outcome x_n, where natural measures are the 0–1 loss, i.e., a zero loss for correct prediction and a unit loss for an error, and the square-error measure. More generally, the predictor can assign probabilities ("soft decision") to the two possible values of the next outcome, making the prediction $b_n = b_n\left(\cdot|x_1^{n-1}\right)$ a conditional probability assignment for x_n given x_1^{n-1}. In an important case, upon observing x_n, the performance of b_n is assessed by the *self-information* loss $l(b_n, x_n) = -\log b_n(x_n)$, also referred to as the *log-loss* function in the machine-learning literature. In this case, the universal prediction problem essentially becomes the universal source coding problem.

The accumulated loss of a predictor along a sequence x_1^n is given by

$$\mathcal{L}(b; x_1^n) = E\left(\sum_{i=1}^{n} l(b_i(x_1^{i-1}), x_i)\right)$$

where the expectation here is due to possible randomization in the predictor.

In the *probabilistic setting* of the prediction problem, it is assumed that the sequence $x_1, x_2, \ldots$ is generated by a stochastic source. If the source is known, one can find the best predictor b that attains $\min_b n^{-1} E\mathcal{L}(b, x_1^n)$, where the expectation here is over the random data using the source distribution. This is a nonuniversal predictor, as it depends on the source distribution. Universal prediction in the probabilistic setting considers the case where the source is unknown. In many cases, this source is of a known type with unknown parameters $\theta \in \Theta$, e.g., a Bernoulli (p) source with p unknown. To get a universal predictor we look for a single predictor U whose performance is as close as possible to the performance of the nonuniversal predictor, tuned to the specific source, for all possible sources. We define the expected regret associated with U as

$$R_n(U) = \sup_{\theta \in \Theta} E\left(\frac{\mathcal{L}(U, x_1^n)}{n} - \min_b \frac{\mathcal{L}(b, x_1^n)}{n}\right)$$

where again the expectation here is due to the random data, using the true source probability. Note that U is the same for all possible $\theta \in \Theta$ but b is chosen differently for each $\theta \in \Theta$. When the loss function is the self-information loss, the regret is called *coding redundancy* or simply *redundancy*. Since we look for U that works well for all sources, the goal is to find U for which $R_n(U)$ is as small as possible, desirably vanishing for large n. In many cases, depending on the class Θ, such a goal is possible.

In the *deterministic setting* of the prediction problem there is no assumption of a probabilistic data-generating mechanism and the predicted sequence is simply an arbitrary individual sequence. Let B be a restricted class of predictors, e.g., B can be the class of single-state predictors or the class of predictors that can be realized by FS machines of a bounded order (an explicit definition of FS predictors is provided later). For a given sequence x_1^n, let $b = b(x_1^n) \in B$ be the predictor from the class B that minimizes $\mathcal{L}(b, x_1^n)$. Clearly, this b is not universal as it depends on the sequence x_1^n. As above, we denote by U the universal predictor, and we define the regret in this case as

$$R_n(U) = \max_{x_1^n} \left(\frac{\mathcal{L}(U, x_1^n)}{n} - \min_{b \in B} \frac{\mathcal{L}(b, x_1^n)}{n} \right)$$

which essentially defines by how much the performance of U deviates, in the worst case, from the performance of the predictor b that is tuned to the sequence. Since we look for a single U that works well for all possible sequences, the goal is to find U for which $R_n(U)$ is as small as possible, desirably vanishing for large n. In many cases, depending on the class B, such a goal is possible.

The existence, the structure, and other properties of universal predictors in both settings were mostly obtained by the well-established universal prediction theory. This paper explores the case where the universal predictors are constrained to be FS machines. As noted above, FS predictors and Markov predictors are also used as a reference class in many occasions. For this we now provide an explicit definition of these machines.

An FS machine with K states, i.e., a K-state machine, is defined by an initial state S_0 and a state transition function

$$S_{n+1} = g(S_n, x_n)$$

where x_n is the binary input, S_n is the machine state at time n, and S_n takes values in the finite set $1, 2, \cdots, K$. The prediction rule b, defined above, is solely governed by the current state, i.e.,

$$b_n = f(S_n).$$

In most of the paper $b_n \in [0, 1]$, reflecting the prediction probability that the next outcome x_n is "1," or the probability assigned to the event that the next outcome x_n is "1." In our binary case we then have for the 0–1 loss $l(b, x) = |b - x|$ (i.e., $l(b, x) = b$ if $x = 0$ and $l(b, x) = 1 - b$ if $x = 1$). For the self-information loss, we have $l(b, x) = -\log b$ when $x = 1$, and $l(b, x) = -\log(1 - b)$ for $x = 0$. For the square-error loss, $l(b, x) = (b - x)^2$.

A *deterministic* FS predictor is a predictor as above where the function $g(\cdot, \cdot)$ is deterministic. Note that we allow deterministic machines to make randomized predictions as long as the state transitions are deterministic (for the 0–1 loss, we essentially assume that the predictor randomly predicts "1," with probability b). We also consider in the paper *randomized* FS predictors where the function $g(\cdot, \cdot)$ is stochastic. In most of the paper we also assume that f and g are independent of the time index n, and therefore the corresponding predictors are *time invariant*.

An important special class of FS machines is the class of order-L Markov machines. In these machines, the current state S_n is determined by the previous L input symbols, i.e., $S_n = (x_{n-1}, \ldots, x_{n-L})$. For binary sequences, these machines have $K = 2^L$ states, and in general they have an important feature that the state is observable from the input sequence.

The regret of a K-state predictor U_K in the probabilistic setting is given by

$$R_n(U_K) = \sup_{\theta \in \Theta} E \left(\frac{\mathcal{L}(U_K, x_1^n)}{n} - \min_b \frac{\mathcal{L}(b, x_1^n)}{n} \right) \quad (1)$$

where as above the expectation is over the random data, using the true source probability. The regret of U_K in the deterministic setting is given by

$$R_n(U_K) = \max_{x_1^n} \left(\frac{\mathcal{L}(U_K, x_1^n)}{n} - \min_{b \in B} \frac{\mathcal{L}(b, x_1^n)}{n} \right). \quad (2)$$

This paper is concerned with the performance of the K-state universal predictor as a function of the number of its states K. Thus, we will generally be interested to examine the behavior of AER

$$R(U_K) = \limsup_{n \to \infty} R_n(U_K). \quad (3)$$

We use $\limsup$ since the sequence x_1^n is arbitrary, but we show later that for deterministic K-state predictors the limit with respect to n always exists. As noted earlier, we will be interested in the behavior of $R(U_K)$ as a function of K; actually, we will mostly investigate its limiting dependence on K.

III. Previous Results in the Probabilistic Setting

In this section, we briefly describe the main results of [7] and [17] which consider universal FS predictors in the probabilistic setting, without the proofs and the detailed analysis. The reader is referred to [7], [5], [17], [16] for further consideration.

A. The 0–1 Loss

In the probabilistic setting, it is assumed that the data to be predicted is generated by a stochastic source with unknown parameters. In our case, the observed binary sequence is generated by an unknown Bernoulli (p) source. The optimal nonuniversal predictor would always predict "0" if $p < 1/2$ and always predicts "1" otherwise, yielding an average of $\min(np, n(1-p))$ errors for a sequence of length n. Thus, the AER in this case is

$$R(U_K) = \limsup_{n \to \infty} E \left[\frac{N_e(U_K, x_1^n)}{n} \right] - \min(p, 1-p) \quad (4)$$

where $N_e(U_K, x_1^n)$ denotes the number of errors made by U_K over the sequence x^n and is equal to $\mathcal{L}(U_K, x_1^n)$ for the 0–1 loss function.

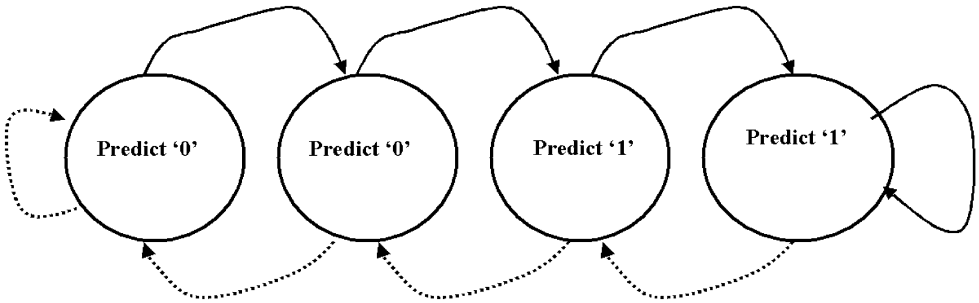

Fig. 1. The four-state SC with a threshold. Solid/dotted lines correspond to transitions generated by "$1/0$."

In [9], it was essentially proved that the AER for this problem (assuming $p \geq \frac{1}{2}$) has the following lower bound for all possible K-state predictors:

$$\forall\, U_K, \qquad R(U_K) \geq \frac{2p-1}{1+\left(\frac{p}{1-p}\right)^{K-1}}. \tag{5}$$

A candidate predictor to attain the optimal performance is the saturated counter (SC) with a threshold. This predictor was proposed and analyzed in [7]. As depicted in Fig. 1, this predictor is composed of a linear array of states, where only adjacent states are connected. The counter is increased/decreased (unless a saturated state is currently occupied) every time a $1/0$ is encountered. It predicts that the next bit will be a "1" if one of the top half states is occupied and "0" otherwise. As shown in [7], it achieves an AER of

$$R(SC_K) = \frac{2p-1}{1+\left(\frac{p}{1-p}\right)^{K/2}}. \tag{6}$$

The work [7] further considers the case of Markov sources, tree sources, and FS sources.

B. The Self-Information Loss

The optimal nonuniversal predictor for the self-information loss, in the probabilistic setting, simply assigns to the next outcome the true source probability distribution $b_n = p(x_n|x^{n-1})$. By doing so the accumulated expected loss is the source entropy. Thus, if the binary sequence is generated by an unknown Bernoulli (p) source, the optimal nonuniversal predictor (source coder) is constant and assigns at each time instant the probability $p_n = p$, thereby achieving the optimal expected code length of $H(p)$ bits per sample, where $H(\cdot)$ is the binary entropy function.

There are many known universal coders that attain the entropy of an unknown Bernoulli source (note the classical work [3]). However, it is interesting to find a universal predictor U_K that is a K-state machine minimizing

$$R(U_K) = \sup_p \limsup_{n\to\infty} E\left[\frac{\mathcal{L}(U_K, x_1^n)}{n} - H(p)\right]. \tag{7}$$

The basic observation made in [17] asserts that a K-state machine can record at most K different probability assignments at any time instant. Consider the Bernoulli (p) case, and suppose that eventually the machine estimates the probability assignment to be q instead of the real value p. This leads to an additional code length of $D(p||q)$ bits per sample over the entropy, where

$$D(p||q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right)$$

is the information divergence. Thus, a basic question that arises is how to quantize optimally the $[0,1]$ probability axis and to allocate these quantized probabilities to the K states. This is a quantization problem with respect to the divergence metric. It turns out that by using a nonuniform grid induced by Jeffreys' prior and for large K, the points $[p_1, \ldots, p_K]$ can be chosen so that

$$\sup_p \min_{i\in\{1,\ldots,K\}} D(p||p_i) \approx \frac{\pi^2}{8\ln(2)K^2}. \tag{8}$$

Clearly, the divergence associated with the best quantizer is a lower bound on the performance of any K-state universal coder. Using this quantization, a K-state *time-variant* machine was constructed in [17], achieving a coding redundancy of $\Theta(\frac{1}{K^2})$, thus being optimal. In this respect, we note that a quantizing problem with respect to (w.r.t.) the divergence was also treated, in a different context, in [13].

As for time-invariant predictors, one can use Pinsker's lemma asserting

$$\forall\, p,q \in [0,1], \qquad D(p||q) \geq \frac{2}{\ln 2}(p-q)^2 \tag{9}$$

and the results in [12] to get a lower bound for the coding redundancy. Specifically, in [12] it was shown that no universal K-state time-invariant machine can achieve a probability estimation whose mean square error with respect to the true probability is smaller than $\Theta\left(\frac{1}{K}\right)$ for all Bernoulli sources.

A *randomized* machine that achieves this bound was also discussed in [12]. This machine uses only K states to estimate the number of ones in a sliding window of length $K-1$, and can be interpreted as an "imaginary sliding window" (see [20]). This machine, called ISW in the sequel, has much of the merits of a true sliding window, see [20]. Following this, it was proved in [17] that when the *randomized* K-state ISW machine is used for universal coding, it attains a coding redundancy of $\Theta(1/K)$. Following [12], it was also shown in [17] that this machine can be derandomized to construct a deterministic K-state machine achieving a coding redundancy of $\Theta\left(\frac{\log K}{K}\right)$.

It turns out, as will be further analyzed in Section V, that the ISW also performs well at the deterministic setting.

IV. Finite Memory Universal Prediction of Individual Sequences for the 0–1 Loss

We turn now to present the novel results of this paper and begin with results on universal K-state prediction of individual sequences w.r.t. the 0–1 loss. We first re-examine the basic result for unconstrained universal prediction w.r.t. the 0–1 loss derived in [6] and suggest a new way to obtain a universal predictor for this setting. We then consider universal prediction using K-state TIDFS machines, where the comparison class is the class of constant predictors. We introduce the LOSC and analyze its performance. A new lower bound is proved which shows that the LOSC is the optimal TIDFS machine for this task. Finally, we generalize the results for the comparison class of L-order Markov machines and general S-state machines.

A. Unconstrained Universal Prediction—Revisited

In the 0–1 loss problem the class of constant predictors contains essentially two predictors—one that predicts "1" with probability one and another that predicts "0" with probability one, as the other predictors are poorer. Therefore, the AER

$$R(U) = \limsup_{n\to\infty} E\left[\frac{N_e(U,x^n)}{n}\right] - \min\left(\frac{n_0}{n}, \frac{n_1}{n}\right) \qquad (10)$$

where n_0 and n_1 are the number of zeros and ones in the sequence and $N_e(\cdot,\cdot)$ is defined as in Section III. In [6], [8], universal predictors with $R(U) = 0$ have been shown. Furthermore, for these predictors, $R_n(U) = O(1/\sqrt{n})$, which is the optimal convergence rate proved by Cover in [1]. The predictor in [6] keeps a count of the number of zeros $n_0(t)$ and the number of ones $n_1(t)$ up to the present time t, then predicts that the next bit will be a "1" with probability $\Phi\left(\frac{n_1(t)}{n_0(t)+n_1(t)}\right)$, where

$$\Phi(x) = \begin{cases} 0, & x < 0.5-\epsilon \\ \frac{x-(0.5-\epsilon)}{2\epsilon}, & 0.5-\epsilon < x < 0.5+\epsilon \\ 1, & x > 0.5+\epsilon \end{cases} \qquad (11)$$

and where $\epsilon = \frac{1}{\sqrt{n_0(t)+n_1(t)}}$. As frequently noted in earlier work, the universal predictor must produce randomized predictions, so that an adversary that knows the predictor cannot generate a sequence on which the predictor always errs.

We now provide another predictor with a different way to assign prediction probabilities. Let

$$A_n = \{y_1^n | n_1(y_1^n) \le n_1(x_1^n)\}.$$

Having observed x_1^n this predictor predicts that the next outcome x_{n+1} is "1" with probability $\frac{|A_n|}{2^n}$. This probability assignment has an intuitive information "weighted mixture" interpretation (see Fig. 2).

It can be proved, following the technique in [6], that this predictor also yields a regret of $R_n(U) = O(1/\sqrt{n})$. The intuitive structure of this predictor may prove to be useful for other settings as well; however, this is left for further research.

Notice that the predictor of [6] and the newly suggested predictor both have a growing number of states (roughly n^2). This leads to the main question of the paper; what happens when the universal predictor is constrained to have only K states?

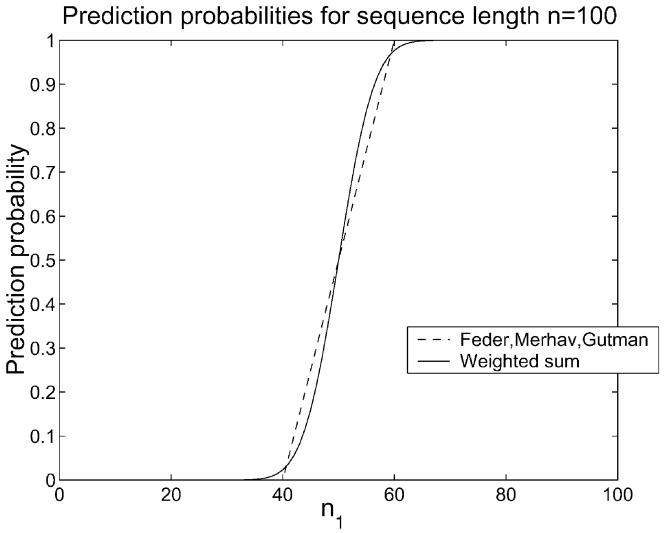

Fig. 2. The predictor in [6] and the weighted mixture predictor.

B. The Linear Output Saturated Counter (LOSC)

We turn now indeed to deal with this question, which is the main subject of the paper. For the 0–1 loss, we focus on the case where this machine is a K-state universal TIDFS machine. The reference class is the class of constant predictors. We introduce the LOSC predictor and analyze its performance. We then prove a lower bound on the performance of any TIDFS predictor as a function of the number of states K. It turns out that no TIDFS can do better than $1/2K$ which is also the asymptotic performance of the LOSC predictor.

The K-state LOSC is composed of a linear array of states, where only adjacent states are connected. The counter is increased/decreased (unless a saturated state is currently occupied) every time a "1/0" is encountered. The LOSC produces randomized predictions—when it occupies the ith state it predicts that the next bit will be a "1" with probability

$$p_i = f(S=i) = \frac{1}{2K} + i \cdot \frac{1}{K}.$$

The LOSC is schematically described in Fig. 3.

Before analyzing the performance of LOSC, we follow [17] and describe a method to find the worst sequence for a given deterministic FS machine.

A deterministic FS machine corresponds to a directed graph where the different states are the vertices and the transitions correspond to edges. Given a certain deterministic FS predictor, each possible binary sequence corresponds to a certain path on its state graph. Define a *minimal cycle* as a cyclical L-element, ordered set of states $\{S_0, S_1, \ldots, S_{L-1}\}$ and input bits $\{x_0, x_1, \ldots, x_{L-1}\}$ such that $\forall\, t \neq \tau$, $S_t \neq S_\tau$ and

$$g(S_0, x_0) = S_1$$
$$g(S_1, x_1) = S_2 \ \ldots g(S_{L-1}, x_{L-1}) = S_0.$$

Given a minimal cycle c, let

$$\mathrm{ref}(c) = \min\left(\frac{\sum_{t=0}^{L-1} x_t}{L}, 1 - \frac{\sum_{t=0}^{L-1} x_t}{L}\right)$$

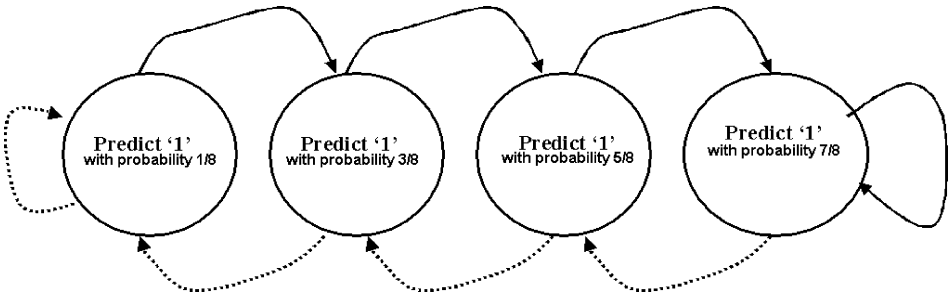

Fig. 3. The LOSC (four states) for the deterministic setting. Solid/dotted lines correspond to transitions generated by "$1/0$."

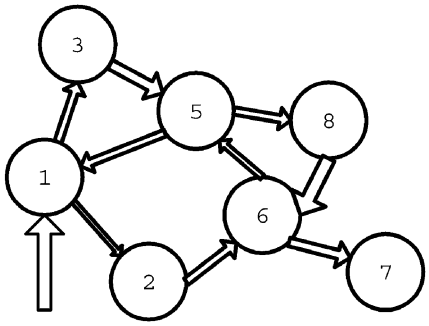

Fig. 4. A state graph consistent with the state sequence below.

be the 0–1 prediction loss obtained by the optimal constant predictor on the cycle sequence. Let

$$P(x_t, S_t) \triangleq \begin{cases} f(S_t), & x_t = 1 \\ 1 - f(S_t), & x_t = 0 \end{cases}$$

be the expected number of correct predictions when the machine occupies state S_t and the input bit is x_t. Thus, the performance obtained by the universal predictor on the cycle sequence is

$$\text{actual}(c) = \frac{\sum_{t=0}^{L-1} P(x_t, S_t)}{L}. \tag{12}$$

The cycle regret is

$$\text{regret}(c) = \text{actual}(c) - \text{ref}(c).$$

Lemma 1: Every path on a state graph of a K-state deterministic FS machine can be broken into minimal cycles and a cycle-free path of size K at most.

Proof: We first show how to trim minimal cycles out of any given sequence. We look for the first occurrence of a repeating state. We take out the minimal cycle that begins and ends with that state and continue this procedure iteratively. For example, the state sequence in Fig. 4 is made up of two cycles and one cycle-free path

$$\begin{aligned} \text{state sequence} &= 1,3,5,1,2,6,5,8,6,7 \\ c_1 &= 1,3,5 \\ c_2 &= 6,5,8,6 \\ \text{path} &= 1,2,6,7. \end{aligned}$$

Clearly, the remainding path after trimming out the minimal cycles cannot be larger than the number of the states K as it does not contain the same state twice. □

We now analyze a path c, associated with a sequence of length L that is composed of two cycles, c_0 and c_1, of lengths L_0 and L_1, respectively. A convexity lemma is proved for the regret of such a sequence.

Lemma 2: The regret of any FS predictor on a sequence, whose path is composed of two minimal cycles, is less than or equal to the weighted regret of the two cycles.

Proof: From the linearity in (12) we obtain

$$\text{actual}(c) = \frac{1}{L}\left(L_1 \cdot \text{actual}(c_1) + L_2 \cdot \text{actual}(c_2)\right).$$

The reference performance for c is

$$\begin{aligned} \frac{1}{L}\min\left(\sum_{x_i \in c, i=0}^{L-1} x_i, L - \sum_{x_i \in c, i=0}^{L-1} x_i\right) \\ \geq \frac{1}{L}\left(L_1 \cdot \text{reference}(c_1) + L_2 \cdot \text{reference}(c_2)\right) \end{aligned}$$

where equality holds iff the same reference predictor is used for both cycles. Thus,

$$\text{regret}(c) \leq \frac{1}{L}\left(L_1 \cdot \text{regret}(c_1) + L_2 \cdot \text{regret}(c_2)\right)$$

and the lemma is proved. □

Note that Lemma 2 is true for all loss functions considered in this paper since the basic requirement is that the optimal reference performance[2] is concave w.r.t. the empirical distribution of the sequence which is always the case. Clearly, it can also be generalized to the case where there are more than two cycles. Another consequence of this lemma is that for deterministic machines and bounded loss functions, the limit in the definition of the regret ((2) and (1)) exists and so we can use $\lim$ instead of $\limsup$. This is due to the fact that for a given sequence length, the maximal regret will be attained by repeating a specific cycle and possibly an extra path whose length is less than K and so the regret at different sequence lengths, n, will be the same up to terms of order $O(\frac{K}{n})$.

Theorem 1: The LOSC with K states achieves an AER of $\frac{1}{2K}$.

Proof: From Lemma 1, each sequence traversing the LOSC is composed of cycles on its state graph and possibly an additional path whose length is at most K. By Lemma 2, in searching for the worst possible sequence for the LOSC, one should look for the cycle with the greatest expected regret. Now, the LOSC has $K-1$ cycles of length 2 between adjacent

[2]Also known as the Bayes envelope.

states, each generated by the input bits "1" followed by "0" (or *vice versa*). The expected number of correct predictions made by the LOSC for each of these cycles is

$$p_i + (1 - p_{i+1}) = \left(\frac{1}{2K} + i \cdot \frac{1}{K}\right) + \left(1 - \left(\frac{1}{2K} + (i+1) \cdot \frac{1}{K}\right)\right) = 1 - \frac{1}{K}. \quad (13)$$

Since each of the reference predictors has exactly one correct prediction in each of these cycles and the cycle length is two, the expected regret of each cycle is $\frac{1}{2K}$. There are two additional possible minimal cycles located at the saturated states. Direct calculation shows that these cycles also yield the same expected regret of $\frac{1}{2K}$. Clearly, a sequence that repeats one of these cycles incurs an extra loss that is between $\frac{1}{2K} - \frac{1}{n}$ and $\frac{1}{2K} + \frac{1}{n}$, and so $R(\mathrm{LOSC}_K) \geq \frac{1}{2K}$.

As noted earlier, the worst case sequence is a repetition of the worst case cycle (for the LOSC all cycles are the same) and an extra path whose length is at most K. Even if the extra path inflicts K errors, the expected regret is still bounded by

$$R_n(LOSC_K) \leq \frac{1}{2K} + O\left(\frac{K}{n}\right) \Longrightarrow R(LOSC_K) = \frac{1}{2K}. \quad (14)$$

Thus, the theorem is proved. □

While our main intent is to analyze the performance of the LOSC for fixed K, one may ask what is the optimal memory size for a sequence of length n assuming we allow K to grow with n. A trivial consequence of the proof of Theorem 1 is that the number of states should be roughly $\sqrt{n}$. This leads to an expected regret of $\frac{1}{\sqrt{n}}$ which is also the optimal convergence rate achieved by the scheme proposed in [6]. The LOSC achieves this performance with a much smaller number of states ($\sqrt{n}$ compared with n^2), but it requires to know n in advance.

Next we show that the LOSC is actually the optimal TIDFS predictor.

Theorem 2: There is no K-state TIDFS machine that achieves an AER smaller than $\frac{1}{2K}$ with respect to the class of constant predictors.

Before proving Theorem 2, we make the following definitions in considering a general (not necessarily a counter) K-state TIDFS predictor. Let

$$p_1 = f(S=1) \leq p_2 = f(S=2) \leq \cdots \leq p_K = f(S=K)$$

be the prediction probabilities assigned by this predictor at the ith state, sorted according to their values. Define the *gap* associated with the predictor as $\max_i(p_{i+1} - p_i)$, i.e., the largest subinterval (excluding the edges) on the probability axis with no states in it. The following lemma holds.

Lemma 3: If the AER of a K-state TIDFS predictor is less than $\frac{1}{2K}$, its gap must be greater than $1/K$.

Proof: A K-state TIDFS predictor with an AER less than $\frac{1}{2K}$ must have $p_1 < 1/2K$ since otherwise the all-zeros sequence will incur an AER that is greater than or equal to $1/2K$. Similarly, $p_K > 1 - 1/2K$ since otherwise the all-ones sequence will incur a greater or equal AER. Thus, for this predictor,

$$p_K - p_1 > 1 - 1/2K - 1/2K = \frac{K-1}{K}.$$

Now, there are $K-1$ consecutive pairs. If Lemma 3 is false, i.e., if $p_{i+1} - p_i \leq 1/K$ for all i, then $p_K - p_1 \leq \frac{K-1}{K}$, yielding a contradiction. □

We now prove Theorem 2.

Proof: Given a K-state TIDFS predictor whose gap is greater than $\frac{1}{K}$, we describe how to construct a sequence yielding an AER greater than or equal to $\frac{1}{2K}$. Clearly, we should consider only predictors whose gap is greater than $\frac{1}{K}$, since Lemma 3 has already shown that the AER of all other predictors is at least $\frac{1}{2K}$.

The sequence is constructed as follows.

1) Start at an arbitrary state.
2) Repeat: If the current prediction probability is above the gap generate the next bit to be "0." Otherwise, generate a next bit "1."

This construction of a "gap-toggling" sequence will also be used later in the paper in proving additional lower bounds.

The expected number of correct predictions made by the predictor for this sequence, denoted Π, is

$$\Pi = \sum_{i=0}^{n_1} p_i^L + \sum_{j=0}^{n_0} \left(1 - p_j^H\right) \quad (15)$$

where the first summation corresponds to the states followed by an incoming bit "1," and the second summation corresponds to the states followed by an incoming bit "0." We denote the probabilities in the first summation by p_i^L since, by construction, these probabilities are beneath the gap. Similarly, we denote the probabilities in the second summation that are above the gap by p_j^H. Without loss of generality we assume $n_1 \geq n_0$, and so the reference predictor will have n_1 correct predictions. Rearranging terms

$$\Pi = \sum_{j=0}^{n_0} \left(1 - (p_j^H - p_j^L)\right) + \sum_{j=n_0+1}^{n_1} p_j^L. \quad (16)$$

By construction, $\left(p_j^H - p_j^L\right) > \frac{1}{K}$ and $p_j^L < 1 - \frac{1}{K}$. Thus,

$$\Pi < n_0 \cdot \left(1 - \frac{1}{K}\right) + (n_1 - n_0) \cdot \left(1 - \frac{1}{K}\right) < n_1 \cdot \left(1 - \frac{1}{K}\right). \quad (17)$$

The expected regret over this sequence is then

$$R = \frac{n_1 - \Pi}{n} > \frac{n_1 - n_1(1 - \frac{1}{K})}{n} = \frac{n_1}{n \cdot K} \geq \frac{1}{2K}, \quad (18)$$

and so the theorem is proved. □

An immediate consequence of Theorem 2 is that the LOSC described above is optimal up to terms that vanish with the sequence length n.

Another way to prove Theorem 2 is to consider a reference predictor that constantly predicts "1" with a probability whose value is the center of the gap. For the gap-toggling sequence,

this predictor predicts the next outcome with a greater probability than any TIDFS machine, thus inducing a regret which is at least half the size of the gap, i.e., at least $1/2K$. This implies that given a gap in the values of the predicting probabilities of a certain machine, it is possible to ensure a regret with respect to the reference predictor located at the center of the gap. Since the considered constant predictor is poorer than the optimal (sequence-dependent) predictor, the regret can only be larger. This technique will serve us in proving the lower bound for the class of Markov predictors.

C. The Class of Order-L Markov Predictors

We now try to find the best K-state TIDFS universal predictor that competes with the class of all order-L Markov predictors.

Theorem 3: There exists a universal predictor with K states that achieves an AER of

$$\left(2\left\lfloor \sqrt[2^L]{\frac{K}{2^L}}\right\rfloor\right)^{-1} = O\left(\left(\sqrt[2^L]{K}\right)^{-1}\right)$$

with respect to the class of order-L Markov predictors (where L is a constant and the result is asymptotic in K).

Proof: We construct a predictor by merging 2^L LOSCs, competing with 2^L constant predictors, each associated with a unique Markov state defined by the previous L symbols. Suppose we allocate D states to each of these LOSCs, and an additional 2^L states to record the current suffix. In the state diagram of the universal predictor we must choose the states to be a Cartesian product of these predictors. Thus, we get a predictor with K states, where

$$K = D^{2^L}2^L \Rightarrow D = \sqrt[2^L]{\frac{K}{2^L}} \geq \frac{1}{\sqrt{2}}\sqrt[2^L]{K}. \tag{19}$$

As shown above, each of the LOSCs achieves an AER bounded by $\frac{1}{2D}$ with respect to best constant predictor. Since the total regret is the weighted average of the constant predictors, the proof is completed. □

We now consider the case where the comparison class is all FS machines with S states. In [6] it was shown that

$$\mu_L(x_1^n) \leq \pi_S(x_1^n) + \sqrt{\frac{\ln S}{2(L+1)}} \tag{20}$$

where $\mu_L(x_1^n)$ and $\pi_S(x_1^n)$ are the fraction of errors obtained by the optimal order-L Markov predictor and the optimal S-state machine, respectively, tuned to the sequence. Utilizing (20) and Theorem 3 we derive the following corollary.

Corollary 1: There exists a universal K-state predictor achieving an AER of at most

$$O\left[\min_L\left(\frac{1}{\sqrt[2^L]{K}} + \sqrt{\frac{\ln S}{2(L+1)}}\right)\right] \tag{21}$$

with respect to the class of all S-state predictors.

Is the predictor proposed in Theorem 3 optimal? Can an adaptive predictor, that counts the number each L-suffix has appeared and dynamically allocates states accordingly, do better? We provide a lower bound on the AER for the class of Markov predictors that has the same asymptotic dependency on K.

Theorem 4: No universal K-state TIDFS predictor achieves an AER less than

$$\left(2^{L+2}\left\lceil \sqrt[2^L]{K}\right\rceil\right)^{-1}$$

w.r.t. the class of all order-L Markov predictors.

The proof of Theorem 4 uses a nontrivial yet tedious vector generalization of the *gap* technique used in the proof of Theorem 2 and is given in Appendix I.

V. Finite Memory Universal Coding and Estimation of Individual Sequences

In prediction with self-information loss the task of the predictor is to assign at each time instant a probability p_t that the next outcome is "1." The associated loss with this assignment and the resulting outcome is the "code length," i.e., $-\log(p_t)$ in case the next bit is actually "1" and $-\log(1-p_t)$ in case the next bit is "0." This universal prediction problem is essentially the universal lossless coding problem as by using arithmetic coding one can turn the probability assigned to the sequence symbols into a codeword of a suitable length (see [19]). Unlike the probability assignment carried out by the universal predictor in Section IV, whose goal is to deal with a malicious adversary, here the probability assignment is the principal task. While we emphasize the self-information loss, the results and theorems in this section hold for both the square-error loss function and the self-information loss unless specified otherwise.

A. Competing With the Class of Constant Predictors—Candidate Machines

Consider the comparison class of single-state predictors, i.e., predictors assigning a constant $p_t = p$. The best static predictor tuned to the sequence predicts the next bit to be "1" with $p = P_{\text{emp}}(x_1^n) = \frac{n_1}{n}$, resulting in a reference code length which is the empirical binary entropy of the sequence. This leads to the definition of the universal coding redundancy with respect to the class of constant predictors

$$R_n(U) = \max_{x^n}\left(\frac{\mathcal{L}(U, x^n)}{n} - H_{\text{emp}}(x^n)\right) \tag{22}$$

where $H_{\text{emp}}(\ ) = H\left(\frac{n_1}{n}\right)$ is the binary empirical entropy of a sequence.

Shtarkov [21] proposed the optimal universal probability assignment that attains the minimal $R_n(U)$, but that solution turns out to be nonsequential in nature as it needs to know n in advance. Alternatively, one can use the Krichevsky–Trofimov probability estimates

$$p_t = \frac{n_1(t) + 0.5}{n_1(t) + n_0(t) + 1}, \qquad \text{for all } 0 \leq t \leq n$$

(see [11]), which are sequential but require unbounded memory. Both schemes achieve a coding redundancy of $\frac{\log n}{2n} + O(1/n)$, where n is the length of the sequence. A different scheme that achieves a vanishing coding redundancy is the Ziv–Lempel ([23]) coding scheme. While this scheme does not explicitly assign a probability to the next bit, it can be translated to a

probability assignment scheme as explained in [6]. This scheme uses unbounded memory as well.

Following the 0–1 loss setting, a natural candidate for a K-state machine is the LOSC. It turns out, however, that it performs poorly in the coding problem. Specifically, the worst case sequence for the LOSC with uniform assigned probabilities is the sequence that begins with $K/2$ ones and continues with the infinite sequence $010101010\ldots$. Thus, it will keep cycling the topmost cycle in its state graph. The average loss attained by this cycle is

$$\frac{-\log(1/2K)-\log(1-3/2K)}{2}\approx\frac{\log 2K}{2},\qquad \text{for } K\gg 1.$$

On the other hand, the empirical entropy of this sequence approaches 1. Thus, the redundancy (regret) grows with K. Modifying the probabilities assigned to the counter states will not help since the counter does not have a "relaxation" feature, meaning that a sequence which starts with consecutive ones or zeros and continues with a balanced sequence will cause the counter to forever toggle around some probability value q that does not match the sequence's empirical probability (which is close to 0.5), resulting in a large coding redundancy.

Another candidate machine is the finite window predictor. This machine keeps track of the last M bits which requires $K=2^M$ states. Suppose even that keeping track of the number of ones in the window can be done with K states (this cannot be done exactly by a K-state machine but a machine with approximately $K=M^2$ states that approximates this count is given in [2]). Now, at each instant the finite window will assign the Krichevsky–Trofimov probability estimate, $\frac{i+0.5}{M+1}$, associated with the counts within a window of size M, where i is the number of ones in the window. The worst case sequence for this finite window predictor toggles between M consecutive ones and M consecutive zeros. Its normalized cumulative self-information loss would be

$$-\sum_{i=0}^{M-1}\frac{1}{M}\cdot\log\left(\frac{i+0.5}{M+1}\right)\approx\int_0^1-\log x dx=\log e$$

leading to a normalized code length that approaches $\log e\approx 1.44$ as M (and thus K) becomes large. The empirical entropy of this sequence approaches 1. Thus, for this sequence, the redundancy of the ideal M-window machine does not diminish with K. A similar result applies for the square-error loss or for any other convex loss function. In general, the finite window predicts poorly any sequence with "nonstationary" behavior.

The best known TIDFS machine so far was given in [17]. This machine is a counter with "reset," as shown in Fig. 5. This machine counts the zeros and ones and uses the Krichevsky–Trofimov probability estimates for prediction. It resets itself roughly every $\sqrt{K}$ steps. As shown in [17], it achieves an asymptotic coding redundancy of $\frac{\log K}{\sqrt{K}}$ for all sequences. Can we do better?

B. The "Exponentially Decaying Memory" (EDM) Machine

We now present a novel K-state TIDFS machine that outperforms the "counter with reset" above, and so it is the best machine known so far. The motivation for this machine is to simulate an exponentially decaying memory over the past data

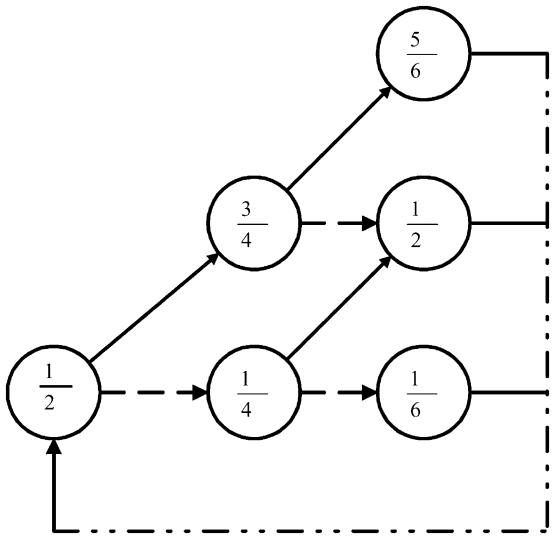

Fig. 5. A counter with reset for $K=6$.

$$f(S_n)=\left(\frac{\sum_{i=0}^{i=n}(1-\lambda)^{n-i}x_i}{\frac{1}{\lambda}}\right)\approx((1-\lambda)f(S_{n-1})+\lambda x_n),\qquad \lambda\in(0,1)$$

where $f(S_n)$, a real number between 0 and 1, is the probability assigned by the machine at time n. A precise implementation of a machine with an exponentially decaying memory as above cannot be done with a finite number of states. Thus, it has to be approximately simulated by a K-state machine, yielding an additional quantization error. Such a machine, called the exponentially decaying memory (EDM) below, is now described and analyzed.

The probabilities assigned to the K states of the EDM are based on a nonuniform quantization of the probability axis in the interval $[K^{-2/3},1-K^{-2/3}]$ so that the density of states assigning a probability in the vicinity of p is proportional to $\frac{1}{(p(1-p))^{1/2+\epsilon}}$ (this choice is motivated by Jeffreys' prior and follows [17]). Note that assuming that the density of states in the vicinity of $p=\frac{1}{2}$ is $\Theta(K)\ \frac{\text{states}}{\text{unit length}}$, the total number of states will be $\Theta(K)$ since $\frac{1}{(p(1-p))^{1/2+\epsilon}}$ is integrable on $[0,1]$ (assuming $\epsilon<0.5$). Choosing a small ϵ will result in fewer states.

Suppose that at time n the machine is at a state S_n with assigned probability $p_n=f(S_n)$. Denote

$$\widehat{p_{n+1}}=p_n(1-K^{-2/3})+x_nK^{-2/3}.$$

The machine's next state function is such that S_{n+1} will be the state whose assigned probability is the closest to $\widehat{p_{n+1}}$ and is between $\frac{1}{2}$ and $\widehat{p_{n+1}}$.

Theorem 5: The EDM achieves an asymptotic coding redundancy of $O\left(K^{-2/3}\right)$ with respect to the empirical entropy.

We provide preliminary definitions and show some properties of the EDM, before proving Theorem 5.

Suppose we order the states according to the prediction probabilities they assign. As the machine moves between states, it may skip over several "state gaps" (for K states there are $K-1$ state gaps). We divide uniformly the loss (code length) obtained at each step, between the state gaps that were skipped over during the transition. For example, suppose the current state assigns a probability of 0.25, the incoming next bit is "1," and suppose that this induces a transition that jumps 10 states upwards. The resulting loss (code length) is 2 bits $(-\log 0.25)$, which is divided between the 10 state gaps, so that each is associated with a loss of 0.2 bits and a $1/10$ up-step (a code length is also accumulated at the extreme states). We denote the size of an x-centered state gap by

$$\Delta(x) = \frac{(x(1-x))^{1/2+\epsilon}}{2K}$$

(see Fig. 6). We assume that all sequences begin and terminate whence the EDM machine occupies the same state. We later show that this assumption is not necessary and Theorem 5 is true for all sequences. By this assumption, the number of times each state gap is crossed on the way up equals the number of times it is crossed on the way down.

An x-centered state gap has a certain accumulated number of up-step, $S_U(x)$, down-steps $S_D(x)$, and an accumulated code length $\mathcal{L}(x)$. We assume $x \leq \frac{1}{2} - \frac{1}{K^{2/3}}$ (the proof for $x \geq \frac{1}{2} + \frac{1}{K^{2/3}}$ follows similarly) and show the following.

1) Up-steps and down-steps that skip over an x-centered state gap could originate only from states within a bounded interval.
2) The ratio between the number of up-steps and total steps that took place over an x-centered state gap is close to x.
3) The average code length, which is the accumulated loss divided by the total number of steps, for each of the state gaps is close to $H(x)$.

Lemma 4: Up-steps that skip over an x-centered gap originate from states in the interval

$$\left[\max\left(K^{-2/3}, \frac{x-\Delta x - K^{-2/3}}{1-K^{-2/3}}\right), x\right].$$

Proof: Assume x_{low} is the lowest state enabling the machine to skip over an x-centered state gap. Therefore,

$$x_{\text{low}} + (1-x_{\text{low}})K^{-2/3} = x - \Delta x. \tag{23}$$

Note that x is a value that lies between states and x_{low} is a value of a state. As a result

$$x_{\text{low}} \geq \max\left(K^{-2/3}, \frac{x-\Delta x - K^{-2/3}}{1-K^{-2/3}}\right).$$

Similarly (see Fig. 7)

$$x_{\text{high}} \leq \frac{x-\Delta x}{1-K^{-2/3}}. \qquad \square$$

Let $\xi_{UD}(x) = \frac{S_U(x)}{S_T(x)}$ where $S_T(x) = S_U(x) + S_D(x)$. The following lemma holds.

Lemma 5:

$$x - O(\Delta(x))K^{2/3} - xO(K^{-2/3}) \leq \xi_{UD}(x) \leq x + xO(K^{-2/3}). \tag{24}$$

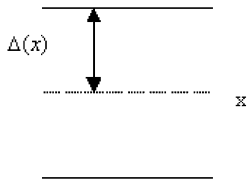

Fig. 6. An x-centered state gap.

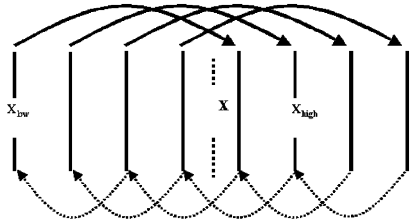

Fig. 7. An x-centered gap, x_{high} and x_{low}. The fact that the length of each up-step is twice the length of each down-step implies that x is in the vicinity of $\frac{1}{3}$.

We prove Lemma 5 in Appendix II.

Note that since $xO(K^{-2/3}) = O\left(\frac{(x(1-x))^{1/2+\epsilon}}{K^{1/3}}\right)$, Lemma 5 implies that

$$|x - \xi_{UD}| \leq O\left(\frac{(x(1-x))^{1/2+\epsilon}}{K^{1/3}}\right).$$

Lemma 6: The normalized code length $\overline{\mathcal{L}}(x) = \frac{\mathcal{L}(x)}{S_T(x)}$ for each state gap obeys

$$\overline{\mathcal{L}}(x) = H(x) + O(K^{-2/3}). \tag{25}$$

Proof: By Lemma 4 for

$$x \notin [K^{-2/3}, 2K^{-2/3}] \cup [1-2K^{-2/3}, 1-K^{-2/3}]$$

each up-step contributes a code length of at most

$$-\log\left(\frac{x-K^{-2/3}}{1-K^{-2/3}}\right).$$

Using Lemma 5 and similar calculations for the down-steps we get

$$\overline{\mathcal{L}}(x) \leq \xi_{UD}(x) \cdot -\log\left(\frac{x-K^{-2/3}}{1-K^{-2/3}}\right) + (1-\xi_{UD}(x)) \cdot -\log\left(1-\frac{x}{1-K^{-2/3}}\right).$$

Since $-\log x > -\log(1-x)$, maximizing the code length is equivalent to maximizing $\xi_{UD}(x)$. Thus, the average code length for each state gap is bounded by

$$\begin{aligned}\overline{\mathcal{L}}(x) &\leq [x + xO(K^{-2/3})] \cdot -\log\left(\frac{x-K^{-2/3}}{1-K^{-2/3}}\right) + [1-x] \cdot \left[-\log\left(1-\frac{x}{1-K^{-2/3}}\right)\right] \\ &\leq H(x) + O(K^{-2/3}).\end{aligned}$$

For $x \in [K^{-2/3}, 2K^{-2/3}]$

$$\begin{aligned}\overline{\mathcal{L}}(x) &\leq \xi_{UD}(x) \cdot -\log\left(K^{-2/3}\right) + (1-\xi_{UD}(x)) \cdot -\log\left(1-2K^{-2/3}\right) \\ &\leq H(x) + O(K^{-2/3}).\end{aligned}$$

$\square$

The derivation for the case $\frac{1}{2}-\frac{1}{K^{2/3}} < x < \frac{1}{2}+\frac{1}{K^{2/3}}$ follows similar algebraic manipulations and leads to the same results.

Summarizing the above, we can now prove Theorem 5.

Proof: In Lemma 5, we saw that the empirical step ratio in an x-centered gap is close to x. Therefore, the optimal average code length for each state gap is at most

$$D\left(x \| x \pm O\left(\frac{(x(1-x))^{1/2+\epsilon}}{K^{1/3}}\right)\right)$$

bits away from $H(x)$. In Lemma 6, we saw that the average code length attributed to each x-centered state gap is also close to $H(x)$. Using the fine quantization approximation for the divergence

$$D(p\|p \pm \delta) = O\left(\frac{\delta^2}{p(1-p)}\right), \qquad \delta \ll p \tag{26}$$

we get that the average code length in each state gap is $O(K^{-2/3})$ away from optimal. The loss per step accumulated at the extreme states is

$$-\log(1-K^{-2/3}) = O(K^{-2/3}).$$

We now use similar arguments to those used in Lemma 2 and show that the redundancy of the entire sequence is bounded by the redundancy of each x-centered gap. The code length obtained by the EDM machine for the entire sequence is the sum of the code lengths obtained at the different state gaps and at the extreme states. Using Jensen's inequality and the concavity of $H(x)$ we get that the code length obtained for the entire sequence by the optimal constant predictor is greater than that of the sum of code lengths obtained by the optimal constant predictors for each gap. Thus, the redundancy for the entire sequence is upper-bounded by the redundancy of the gaps, i.e., $O(K^{-2/3})$.

To generalize the proof for sequences that do not begin and end at the same state, we note that the initial state can always be reached by no more than K steps. These K steps can inflict a loss that is bounded by $K\log(K^{2/3})$ which affects the coding redundancy by no more than $\frac{K\log(K^{2/3})}{n}$ which does not affect the asymptotic coding redundancy. □

When using exponentially decaying memory machines (even with an unbounded number of states) one would like the exponential decay factor λ to be as small as possible to enable accurate weighting of a distant past. The resulting loss is proportional to λ. However, a small decay factor increases the effective required memory and results in a larger quantization error when simulated by an FS machine. The error is proportional to $1/(\lambda K)^2$. From this we see that the optimal tradeoff is attained when $\lambda = K^{-2/3}$, as the loss in reducing the effective memory equals the loss that incurs due to quantization error, i.e., $O(K^{-2/3})$. This is the optimal loss obtained by the EDM, as shown in Theorem 5.

C. Lower Bound

In the probabilistic setting there exists a lower bound for our problem of $\Theta(\frac{1}{K})$ as a consequence of the lower bound shown in [12]. Clearly, this bound also applies in the deterministic setting. We now prove a stronger bound of $\Theta(\frac{1}{K^{4/5}})$, which demonstrates that the deterministic setting is fundamentally different from the probabilistic setting. Note that for unbounded memory both settings exhibit a $\frac{\log n}{2n}$ behavior.

Theorem 6: Any TIDFS machine whose coding redundancy for all sequences is less than $\frac{1}{4M}$ must have at least $\Theta(M^{5/4})$ states.

Before we present the formal proof of Theorem 6 we provide its outline. For a given machine, we construct sequences called $T(x)$ (stands for "threshold sequence—x"), indexed by a value $x \in [0,1]$ as follows. If at time n the machine is at some state whose probability assignment is greater (or equal) than x the corresponding value of $T(x)$ is "0"; otherwise its value is "1." These sequences have some interesting properties. Each $T(x)$ circles in a cycle of states (after at most K steps) since by its construction the next state depends solely on the current state. Another property is shown in Lemma 7, asserting that for a machine whose coding redundancy is less than $\frac{1}{4M}$, the majority of the states associated with $T(x)$ must have probabilities within $\Theta(\frac{1}{2M})$ of the value x. Furthermore, for such a machine we show that the frequency of ones in the cycle should also be in the vicinity of x, and so the cycle length should be large enough, typically of the size $\Theta(M^{1/4})$, to provide the necessary precision. Following this we consider M such sequences, indexed by M uniformly spaced values of x. The overlap between the states of the cycles associated with each sequence is small. The machine has to deal with all these sequences and so its total number of states should be at least the size of the union of these cycles, which is $\Theta(M^{5/4})$ states.

We now prove the following lemma showing the properties of $T(x)$ previously discussed.

Lemma 7: If a TIDFS machine achieves a coding redundancy of $\frac{1}{4M}$ then the probabilities associated with at least half of the states belonging to the sequence $T(x)$ are within $\frac{1}{2M}$ of x. Furthermore, the empirical probability of $T(x)$ is within $\frac{1}{2\sqrt{M}}$ of x.

Proof: The average code length Λ, attained by a machine for each $T(x)$, is

$$\Lambda = \frac{1}{L}\left(\sum_{i=0}^{L_1-1} -\log\left(p_i^L\right) + \sum_{j=0}^{L_0-1} -\log\left(1-p_j^H\right)\right) \tag{27}$$

where $L = L_0 + L_1$ is the cycle length and L_0 and L_1 are the number of zeros and ones in the cycle generated by $T(x)$. By construction

$$\forall i,j : p_i^L \le x \le p_j^H.$$

Clearly, replacing each of the probabilities with x would reduce the code length. The derivative of $\log_2(x)$ is larger than 1 in the interval $(0,1)$. Therefore, if half of the probabilities of the states associated with $T(x)$ are more than $\frac{1}{2M}$ away from x, the coding redundancy with respect to a constant predictor that always predicts x is larger than $\frac{1}{4M}$, thereby contradicting the assumption on the coding redundancy. This proves the first claim of the lemma.

As for the second claim, from above the average code length obtained by the machine for each $T(x)$ is larger than $-\frac{L_1}{L}\log(x) - \frac{L_0}{L}\log(1-x)$. On the other hand, the code

length achieved by the optimal constant predictor for that sequence is $H(\frac{L_1}{L})$. Therefore, the coding redundancy of the sequence is larger than $D(\frac{L_1}{L}\|x)$ which, in turn, by Pinsker's inequality, is larger than $(\frac{L_1}{L}-x)^2$. Since, by assumption, the coding redundancy is less than $\frac{1}{4M}$ we get

$$\left(\frac{L_1}{L}-x\right)^2 \le \frac{1}{4M} \Longrightarrow \left|\frac{L_1}{L}-x\right| \le \frac{1}{2\sqrt{M}}. \quad (28)$$

□

The fact, shown above, that the empirical probability of $T(x)$ is close to x implies that its cycle length L must be large enough so that the rational number $\frac{L_1}{L}$ is in the vicinity of x. This is demonstrated in the following example. Suppose the machine achieves a coding redundancy of 0.01, and consider a sequence $T(0.86)$. Its cycle length should be at least 5, since $B = \{1/4, 1/3, 1/2, 2/3, 3/4\}$, the set of empirical probabilities that can be created by cycles of a lesser length fails to obey (28) as it does not contain any value in the 0.1 neighborhood of 0.86.

We now consider M sequences $\{T(x_i)\}$ associated with M uniformly spaced x-values in the interval $[0, 1]$. As noted earlier, the cycle length of each $T(x)$ must be large enough. It turns out, as shown in Lemma 8, that we can lower-bound S_{CL}, the sum of the lengths of the M cycles associated with all these sequences.

Lemma 8: S_{CL} is at least $\Theta\left(M^{5/4}\right)$ for any TIDFS whose coding redundancy is less than $\frac{1}{4M}$.

Proof: Consider a sequence $T(x)$ and the set $A(x)$ of rational numbers that are within a distance $1/2\sqrt{M}$ of x. We denote by $L_{\min}(x)$ the smallest denominator of these rational numbers. The size of the cycle associated with $T(x)$ is at least $L_{\min}(x)$, since cycles of a smaller length yield rational empirical probabilities whose denominator is smaller than $L_{\min}(x)$ and hence are not in $A(x)$.

Clearly, a rational number can be in the set $A(x)$ of x's that are within $1/2\sqrt{M}$ of it. Thus, a rational number can be in at most $\Theta(\sqrt{M})$ sets associated with the uniformly spaced x_i. As the cycle length of $T(x_i)$ is lower-bounded by the smallest denominator of the rational numbers in the set $A(x_i)$, a lower bound on S_{CL}, the sum of these cycle lengths, is obtained by allocating the rational numbers with the smallest possible denominators to as many possible sets $A(x_i)$. Suppose, indeed, that we order the rational numbers by their denominator size, i.e., $0/1, 1/1, 0/2, 1/2, 2/2, 0/3, 1/3, \ldots$, and assume that according to this order, each is associated with $\Theta(\sqrt{M})$ x-values. Clearly, this assignment leads to a lower bound on S_{CL}, the sum of the cycle lengths. We need at least $\Theta(\sqrt{M})$ different rational numbers to cover the M equally spaced x_i's. Thus, since there are $L+1$ rational numbers with denominator L we need to use all the rational numbers up to denominator K where

$$\sum_{L=1}^{K}(L+1) = \Theta(\sqrt{M})$$

that is, up to $K = \Theta(M^{1/4})$.

To get the lower bound on S_{CL}, we use the assignment above, and note that each rational J/L whose cycle length is L corresponds to $\Theta(\sqrt{M})$ values of x_i, thereby contributing $L\Theta(\sqrt{M})$ to the sum. As previously noted, there are $L+1$ rational numbers in $[0, 1]$ whose denominator is L. Combining the above, we get

$$S_{\mathrm{CL}} \ge \Theta(\sqrt{M})\left(\sum_{L=1}^{\Theta(M^{1/4})} L(L+1)\right) = \Theta(M^{5/4}). \quad (29)$$

□

We can now prove Theorem 6, based on the preceding lemmas.

Proof: The required number of states of a TIDFS machine whose redundancy is smaller than $\frac{1}{4M}$ is lower-bounded by the number of states in the union of the set of states associated with all the x_i's. By Lemma 7, at most half of the states associated with different x_i can overlap. Thus, the size of this union is at least half of S_{CL}. The proof is completed by utilizing Lemma 8 that provided a bound on S_{CL}. □

Note that Theorem 6 implies that any TIDFS machine with K states cannot achieve a coding redundancy of less than $\Theta(K^{-4/5})$.

A universal encoder that competes with the class of order-L Markov encoders is constructed in the same way the universal predictor was constructed in Theorem 3. This construction will yield a coding redundancy of

$$\left(2\left\lfloor \sqrt[2^L]{\frac{K^{2/3}}{2^L}} \right\rfloor\right)^{-1}$$

to the order-L Markov empirical entropy.

D. Randomized Machines and the ISW

A question that arises is whether an optimal randomized machine can reduce the regret for the self-information and square-error loss functions. We now analyze an interesting randomized FS machine, the ISW, and show that it attains better performance than the deterministic machines and for the square-error loss it attains optimal performance.

The ISW, shown in Fig. 8, was previously introduced (see [20], [17], [12]) as an economic way to simulate the K-size window. It uses $K+1$ states to keep track of the number of ones in a sliding window of length K and works as follows. Suppose that at time t the machine occupies the ith state, that is, it estimates the number of ones in the window of the last K bits to be i. If the next bit is "1," the machine state should move up by one or remain at the same state, depending on the value of the bit observed at time $t-K$, that should be removed. However, remembering the last K bits requires 2^K states. The ISW approximates this situation and assumes that the removed bit is "1" with probability $\frac{i}{K}$. Thus, it is a randomized machine

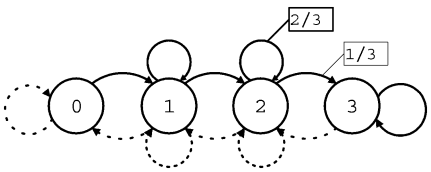

Fig. 8. The four-state ISW. Solid/dotted lines are for transtions generated by "1/0." Transition probabilities are given for the case $i = 2$ and incoming $bit = 1$.

which when it occupies state i and the next bit is "1," it moves up to state $i+1$ with probability $1-\frac{i}{K}$ and remains in the same state with probability $\frac{i}{K}$. The probability assigned to the next bit when the machine occupies state i is $\frac{i}{K}$ in the least square prediction setting and $\frac{i+0.5}{K+1}$ for the universal coding setting.

In [20], it was shown that for Bernouli sources this machine "simulates" and has many features of a true sliding window. For example, for a Bernoulli (p) source, the state that the machine occupies is a binomial random variable, describing the number of ones in the last K experiments. It turns out that the variance of the probability assignment is about $\frac{p(1-p)}{K}$, and the divergence describing the extra loss when the machine assigns the probability q is

$$D(p\|q) \propto \frac{(p-q)^2}{p(1-p)}$$

(in fine uniform quantization). Thus, the *randomized* $K+1$-state ISW machine can be used for universal coding (or least square prediction) of *Bernouli* sequences, attaining a redundancy of $\Theta(1/K)$.

Following this, it is interesting to see whether the ISW will perform as well for deterministic sequences. At first glance it seems the ISW will predict poorly since the machine it is imitating, the finite window, predicts poorly if the sequence does not have a stationary behavior. Nevertheless, as shown later, the ISW achieves a least square prediction redundancy of $\Theta(K^{-1})$, which is the same redundancy it incurs in the probabilistic Bernoulli setting.

We begin the analysis by showing some features of the probability estimates of the ISW. Recall that the probability assignment associated with the ith-state of the ISW is $\frac{i}{K}$. Let p_n be the probability assignment given by the ISW at time n. For an ISW with $K+1$ states this random variable can take the values $\{0, 1/K, \ldots, 1\}$. Let E_n be the expected value of p_n

$$E_n = \sum_{i=0}^{K} \Pr\left(p_n = \frac{i}{K}\right) \cdot \frac{i}{K}.$$

By construction of the ISW, if $x_n = 1$ then

$$\Pr\left(p_{n+1} = \frac{i}{K}\right) = \Pr\left(p_n = \frac{i}{K}\right) \cdot \frac{i}{K} + \Pr\left(p_n = \frac{i-1}{K}\right) \cdot \left(1 - \frac{i-1}{K}\right).$$

Therefore,

$$\begin{aligned} E_{n+1} &= \sum_{i=0}^{K} \Pr\left(p_{n+1} = \frac{i}{K}\right) \cdot \frac{i}{K} \\ &= \sum_{i=0}^{K} \left[\Pr\left(p_n = \frac{i}{K}\right) \cdot \frac{i}{K} + \Pr\left(p_n = \frac{i-1}{K}\right) \cdot \left(1 - \frac{i-1}{K}\right)\right] \cdot \frac{i}{K} \\ &= \sum_{i=0}^{K} \left[\Pr\left(p_n = \frac{i}{K}\right) \cdot \frac{i}{K}\right] \cdot \frac{i}{K} \\ &\quad + \sum_{i=0}^{K} \left[\Pr\left(p_n = \frac{i}{K}\right) \cdot \left(1 - \frac{i}{K}\right)\right] \cdot \frac{i+1}{K} \\ &= \sum_{i=0}^{K} \left[\Pr\left(p_n = \frac{i}{K}\right) \cdot \left(\frac{i}{K} + \left(1 - \frac{i}{K}\right)\right)\right] \cdot \frac{i}{K} \\ &\quad + \sum_{i=0}^{K} \left[\Pr\left(p_n = \frac{i}{K}\right) \cdot \left(1 - \frac{i}{K}\right)\right] \cdot \frac{1}{K} \\ &= E_n \cdot \left(1 - \frac{1}{K}\right) + \frac{1}{K}. \end{aligned}$$

Similarly, it can be shown that if $x_n = 0$, then $E_{n+1} = E_n \cdot (1 - \frac{1}{K})$. Thus, in general

$$E_{n+1} = E_n \cdot \left(1 - \frac{1}{K}\right) + x_n \cdot \frac{1}{K} \tag{30}$$

implying that the sequence of expected probability assignments of the ISW is an exponential decaying memory sequence, with a decaying factor of $1 - \frac{1}{K}$.

We prove in the Appendix III that the variance of the ISW probability assignments

$$V_n = \sum_{i=0}^{K} \Pr\left(p_n = \frac{i}{K}\right) \cdot \left(\frac{i}{K} - E_n\right)^2 \tag{31}$$

satisfies the recursion

$$V_{n+1} = V_n \cdot \left(1 - \frac{2}{K}\right) + \frac{E_n - E_n^2}{K^2}. \tag{32}$$

Lemma 9: There exists $C > 0$ depending only on K such that for all binary sequences

$$V_n \le \frac{C(E_n - E_n^2)}{K}.$$

Proof: We prove Lemma 9 by induction. Assume that the property holds for V_n and that without loss of generality $E_n \ge 0.5$

$$\begin{aligned} V_{n+1} &= V_n \cdot \left(1 - \frac{2}{K}\right) + \frac{E_n - E_n^2}{K^2} \\ &\le \frac{C(E_n - E_n^2)}{K} \cdot \left(1 - \frac{2}{K}\right) + \frac{E_n - E_n^2}{K^2} \\ &= \frac{CA\left(E_{n+1} - E_{n+1}^2\right)}{K} \cdot \left(1 - \frac{2}{K}\right) + \frac{A\left(E_{n+1} - E_{n+1}^2\right)}{K^2} \end{aligned} \tag{33}$$

where $A = \frac{E_n - E_n^2}{E_{n+1} - E_{n+1}^2}$. The lemma holds trivially (for $C \ge 0.5$) if

$$\begin{aligned} E_n - E_n^2 &\ge E_{n+1} - E_{n+1}^2 \\ |E_n - 0.5| &\ge |E_{n+1} - 0.5|. \end{aligned}$$

Otherwise, $|E_n - 0.5| \le |E_{n+1} - 0.5|$. Since $E_n \ge 0.5$, we conclude that $x_n = 1$.

If one could prove that

$$AC\left(1 - \frac{2}{K}\right)K + A \le CK$$

or equivalently

$$A \le \frac{CK}{CK - (2C-1)} = \frac{1}{1 - \frac{2C-1}{CK}} \tag{34}$$

then the lemma would follow from (33). To show (34), we use series expansion

$$E_{n+1} - E_{n+1}^2 = E_n - E_n^2 + (1-2p)\frac{1 - E_n}{K}$$

where $E_n < p < E_{n+1}$ (also noting that $p > 0.5$). Therefore,

$$A = \frac{E_n - E_n^2}{E_n - E_n^2 + (1-2p)\frac{1-E_n}{K}} = \frac{1}{1 - \frac{2p-1}{E_n K}}. \tag{35}$$

For (34) to hold we require that

$$\frac{2p-1}{E_n K} \le \frac{2C-1}{CK}$$
$$\Downarrow$$
$$C \ge \frac{E_n}{1 + 2(E_n - p)}.$$

Thus, it is sufficient to require

$$C = \frac{1}{1 - \frac{2}{K}}. \tag{36}$$

This proves the lemma for $K \ge 3$ and we notice that C approaches 1 as K goes to infinity. □

Theorem 7: The ISW achieves a redundancy of at most $O(\frac{1}{K})$ w.r.t. the squared error loss function.

Proof: The normalized cumulative loss for the ISW, denoted $\mathcal{CL}(x_0^t)$, is

$$\begin{aligned}
\mathcal{CL}(x_0^t) &= \frac{1}{t} E\left[\sum_0^t (x_n - p_n)^2\right] \\
&= \frac{1}{t}\sum_0^t \sum_{i=0}^K \Pr\left(p_n = \frac{i}{K}\right)\left(\frac{i}{K} - x_n\right)^2 \\
&= \frac{1}{t}\sum_0^t \sum_{i=0}^K \Pr\left(p_n = \frac{i}{K}\right) \\
&\quad \cdot \left(\frac{i}{K} - E_n + E_n - x_n\right)^2 \\
&= \frac{1}{t}\sum_0^t \sum_{i=0}^K \Pr\left(p_n = \frac{i}{K}\right) \\
&\quad \cdot \left[\left(\frac{i}{K} - E_n\right)^2 \right. \\
&\qquad \left. + 2\left(\frac{i}{K} - E_n\right)(E_n - x_n) + (E_n - x_n)^2\right].
\end{aligned}$$

Since $\sum_{i=0}^K \Pr\left(p_n = \frac{i}{K}\right)\frac{i}{K} = E_n$, the extra loss (redundancy) is

$$\begin{aligned}
R(x_0^t) = \frac{1}{t}\sum_0^t \sum_{i=0}^K \Pr\left(p_n = \frac{i}{K}\right) \\
\cdot \left[\left(\frac{i}{K} - E_n\right)^2 + (E_n - x_n)^2\right].
\end{aligned} \tag{37}$$

Thus, the redundancy is made up of two terms. The first is the redundancy of an exponential decaying memory machine with a decaying factor of $1 - \frac{1}{K}$. Notice that E_n is real and there is no quantization error. Therefore, following the derivations made in the proof of Theorem 5 for exponential decaying memory machines, it contributes a redundancy of at most $O(\frac{1}{K})$. The second term is bounded by Lemma 9. Specifically

$$\begin{aligned}
\sum \Pr\left(p_n = \frac{i}{K}\right)(E_n - x_n)^2 \\
= V_n\left(\frac{C(E_n - E_n^2)}{K}\right) \le \left(\frac{1}{K}\right).
\end{aligned} \tag{38}$$

Thus, the sum of the two terms is $O(\frac{1}{K})$ and the theorem is proved. □

The preceding discussion reveals an interesting feature of the ISW. The ISW was originally designed to simulate a fixed finite window with perfect memory for a bounded interval. However, it actually has an exponential decaying memory, so that it can track a changing data behavior and be competitive in predicting any individual sequence.

To make the ISW suitable for the coding problem one needs to bound the probabilities away from zero. This can be done by using the Krichevsky–Trofimov probabilities (for the probability assignment but not for the transition probabilities) or by restricting the extreme probability assignments to be $\frac{1}{K}$ and $1 - \frac{1}{K}$ instead of 0 and 1.

We now provide an approximated analysis of the average code length (the normalized cumulative loss)

$$\mathcal{CL}(x_0^t) = \frac{\mathcal{L}(U, x_1^t)}{t}$$

and the redundancy of the ISW. The analysis is approximate since we use the expressions for E_n and V_n derived earlier, while we should have used slightly different expressions associated with the Kriechevsky–Trofimov probability assignments, or the bounded assignments, needed for prediction with log loss

$$\begin{aligned}
\mathcal{CL}(x_0^t) &= \frac{1}{t} E\left[\sum_0^t -\log p(x_n)\right] \\
&= \frac{1}{t}\left[\sum_0^t \sum_{i=0}^K -\Pr\left(p_n = \frac{i}{K}\right)\log\left(1 - \left|x_n - \frac{i}{K}\right|\right)\right] \\
&= \frac{1}{t}\left[\sum_0^t \sum_{i=0}^K -\Pr\left(p_n = \frac{i}{K}\right)\log\left(1 - \left|x_n - \frac{i}{K}\right|\right)\right].
\end{aligned}$$

Dividing the different instances into bins according to the current expectation and using the fact that the expectation is close to the empirical ratio of ones and zeros (following the proof of Theorem 5), we get

$$\begin{aligned}
\mathcal{CL}\left(x_0^t\right) &\approx \frac{1}{t}\left[\sum_0^t \sum_{i=0}^K -\Pr\left(p_n = \frac{i}{K}\right)\left(E_n \log\left(\frac{i}{K}\right)\right.\right. \\
&\qquad \left.\left. + (1 - E_n)\left(\log\left(1 - \frac{i}{K}\right)\right)\right)\right] \\
&= \frac{1}{t}\left[\sum_0^t \sum_{i=0}^K -\Pr\left(p_n = \frac{i}{K}\right)\right. \\
&\qquad \times \left(E_n\left(\log\left(\frac{i}{K}\right) \pm \log(E_n)\right)\right. \\
&\qquad + (1 - E_n)\left(\log\left(1 - \frac{i}{K}\right)\right. \\
&\qquad \left.\left.\left. \pm \log(1 - E_n)\right)\right)\right] \\
&= \frac{1}{t}\sum_0^t \left[-E_n \log(E_n) - (1 - E_n)\log(1 - E_n)\right] \\
&\quad + \frac{1}{t}\left[\sum_0^t \sum_{i=0}^K \Pr\left(p_n = \frac{i}{K}\right) D\left(E_n \| \frac{i}{K}\right)\right].
\end{aligned}$$

Since the ISW's expectation has an exponential memory the redundancy depends on the term

$$\sum_{i=0}^{K} \Pr\left(p_n = \frac{i}{K}\right) D\left(E_n \| \frac{i}{K}\right).$$

Using the fine quantization approximation for the divergence would yield a redundancy of $O(\frac{1}{K})$. However, this approximation can no longer be justified since it is not valid for states whose value is far from E_n. Nevertheless, since we know that the ISW's variance is bounded by $O(\frac{C(E_n - E_n^2)}{K})$ and the probabilities are bounded away from zero it can be proved that the term

$$\sum_{i=0}^{K} \Pr\left(p_n = \frac{i}{K}\right) D\left(E_n \| \frac{i}{K}\right)$$

(the weighted divergence) is bounded by $O(\frac{\log K}{K})$. We need only to consider probability distributions with one, two, or three nonzero probability values (yet obeying the variance constraint). The extreme case is where $E_n = 0.5$ and both sides of the probability axis contribute to the divergence. By standard maximization it is seen that the extreme distribution is the one where 0.5 gets a mass of about $1 - \frac{2}{K}$ and a mass of about $\frac{1}{K}$ is assigned to the values $\frac{1}{K}$ and $1 - \frac{1}{K}$. This yields a weighted divergence of $O(\frac{\log K}{K})$.

We conjecture that by analyzing the higher moments of the ISW's probability estimates, it will be shown that the state distribution is close to binomial. This will enable us to use the fine quantization approximation to get an $O(\frac{1}{K})$ redundancy, for the coding problem.

VI. Conclusion and Summary

The problem of universal prediction of individual sequences when the universal predictor is constrained to be an FS machine has been explored. This problem has a clear practical importance, but it also has important theoretical consequences. In the case of the 0–1 loss function, we have shown that the natural and heuristic algorithms based on an SC are optimal compared to the classes of constant predictors and Markov predictors, i.e., exhibit the same dependency in K, the number of states, as the lower bounds we proved.

For the self-information loss and deterministic machines, we have presented a new finite-memory probability assignment algorithm that performs better than the best algorithms known so far. This machine has effectively an exponentially decaying memory. We have also presented a new lower bound for the probability assignment problem with finite memory. We conjecture that the lower bound is not tight for this problem and actually the optimal way to assign probabilities to the next outcome with finite memory, is by using the exponentially decaying memory machine.

Using common randomness (needed for the decoder) one may also consider randomized machines and use the imaginary sliding window as a coding/estimating machine. We have proved that this machine achieves a coding redundancy of $O(\frac{\log K}{K})$ and a least square regret of $O(\frac{1}{K})$. Note that using a randomized machine for an individual sequence yields the same (or almost the same) regret as for probabilistic sequences. This is not the case for deterministic machines. We conjecture that this result represents a game-theoretic principle (that holds under some regularity conditions). " If a randomized strategy is used in competing against a malicious opponent, this adversary cannot gain more then if it has used a random strategy, thus in effect it is limited." It will be interesting to identify this principle in other settings.

Some of the results in this paper were also generalized for larger alphabets (see [15]). There are still quite a few open problems for further research whose solution will compliment the results of this paper. For example, it will be interesting to find an $o(K^{2/3})$ algorithm for the probability assignment problem using TIDFS machines, or to show that such an algorithm does not exist. Another interesting problem is to solve the general "universal prediction with expert advice," i.e., competing with any M experts and general loss function, using a restricted K-state predictor. On another level, a very interesting problem is to find finite-memory twice-universal predictors, i.e., universal predictors that compete not only with unknown model parameters but also with an unknown model class. These problems and other related problems are currently under investigation.

Appendix I
Proof of Theorem 4

Each state $S_k k = 1, \ldots, K$ will be associated with a state vector of length 2^L of probabilities, denoted by $\bar{S}_k$, corresponding to the prediction probabilities that the machine produces when one of the 2^L suffixes of length L follows the state S_k. Such a vector is described Fig. 9 for the case $L = 2$, where $\bar{S}_k = \langle 0.2, 0.6, 0.5, 0.9 \rangle$. We will describe a vector of gaps (vector gap), $\bar{G}$, of dimension 2^L, where each coordinate of this vector is a subinterval of $[0, 1]$ so that each of the K states has at least one state vector coordinate outside the $\bar{G}$ coordinates (see Fig. 10). We then consider the reference Markov predictor whose assigned probability at each Markov state is located at the subinterval center of the corresponding state of $\bar{G}$.

We now show that $\bar{G}$ with the above prescribed feature indeed exists. We divide the interval $[0, 1]$ into $\lceil \sqrt[2^L]{K} \rceil$ equal intervals of length $\lceil \sqrt[2^L]{K} \rceil^{-1}$. The set $\mathcal{G}$ is made up of all possible vectors (of dimension 2^L) whose coordinates are the subintervals described above and thus its cardinality is $\lceil \sqrt[2^L]{K} \rceil^{2^L} > K$. Since each state vector $\bar{S}_k$ can only account for one possible vector $\bar{g} \in \mathcal{G}$ in the sense that $\forall\, 1 \leq i \leq 2^L$, $\bar{S}_{k_i} \in \bar{g}_i$, there is a least one $\bar{G} \in \mathcal{G}$ with the desired feature (here we assume $\lceil \sqrt[2^L]{K} \rceil > \sqrt[2^L]{K}$; otherwise, replace K with $K + \epsilon\, \epsilon \to 0$ and the same results hold).

As noted above, we consider a reference Markov predictor whose prediction probabilities $f_R(t)$ are located at the center of the coordinates of $\bar{G}$. This will guarantee that at each time instant t we can reach in L steps a state for which the prediction probability assigned by the machine $f_m(t + L)$ and the prediction probability assigned by the reference Markov predictor $f_R(t+L)$ differ by at least $\frac{1}{2} \cdot \lceil \sqrt[2^L]{K} \rceil^{-1}$. We refer to this L-step path as "malicious suffix" and denote the sequence of bits that

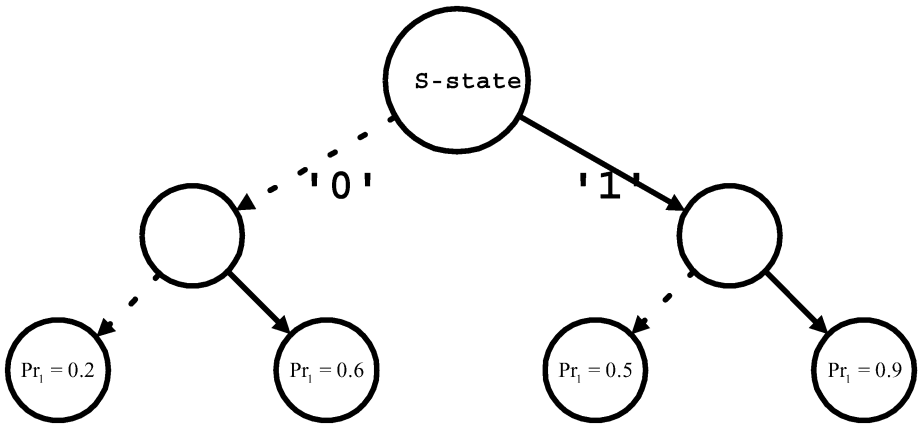

Fig. 9. A state diagram corresponding to the state vector $\bar{S}_k = \langle 0.2, 0.6, 0.5, 0.9 \rangle$.

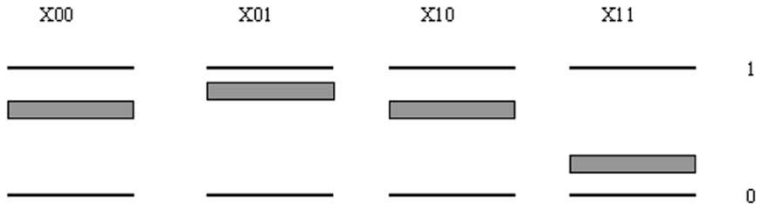

Fig. 10. A vector of gaps for $L = 2$. Each state vector has at least one coordinate outside the vector of gaps.

generate it by $\{MS(i)\}_{i=0}^{i=L-1}$. This means that if the prediction probability of the machine is below (or above) that of the reference Markov predictor, an incoming bit "1" (or "0") will yield a relative loss (or regret) of at least $\frac{1}{2} \cdot \lceil \sqrt[2^L]{K} \rceil^{-1}$. However, it seems that we cannot be certain what will be the relative performance of the two machines (the given machine and the reference Markov predictor) on the L-step path. Nevertheless, it turns out that we can design an algorithm, shown in Fig. 11, that will guarantee an amortized loss for each step.

The algorithm works as follows. At each time instant it checks whether the difference between the prediction probability and the reference probability is above a certain threshold (that depends on the iteration index, i). If we are below the threshold, we move to the next bit of the "malicious suffix" as we are at the limits of the required loss. However, if it is above the threshold, we must introduce a loss and we choose the next bit accordingly. The algorithm stops when either it reached the end of the "malicious suffix" (after at most L steps), or after it accumulated enough loss.

Assume that the algorithm initializes at $i = m$ where $m \leq L - 1$. Then, the regret is bounded by

$$R \geq \frac{1}{m+1} \cdot \frac{2^{m+1} - 1 - \sum_{i=0}^{i=m-1} (2^{i+1} - 1)}{2^{L+1} - 1} \frac{1}{2 \left\lceil \sqrt[2^L]{K} \right\rceil} \tag{39}$$

$$= \frac{1}{m+1} \cdot \frac{m+1}{2^{L+1} - 1} \cdot \frac{1}{2 \left\lceil \sqrt[2^L]{K} \right\rceil} \geq \frac{1}{2^{L+2} \left\lceil \sqrt[2^L]{K} \right\rceil} \tag{40}$$

and the theorem is proved.

Corollary 2: No universal TIDFS predictor with K states achieves an AER less than $\frac{1}{4 \cdot S \lceil \sqrt[S]{K} \rceil}$ w.r.t. the class of all S-state machines.

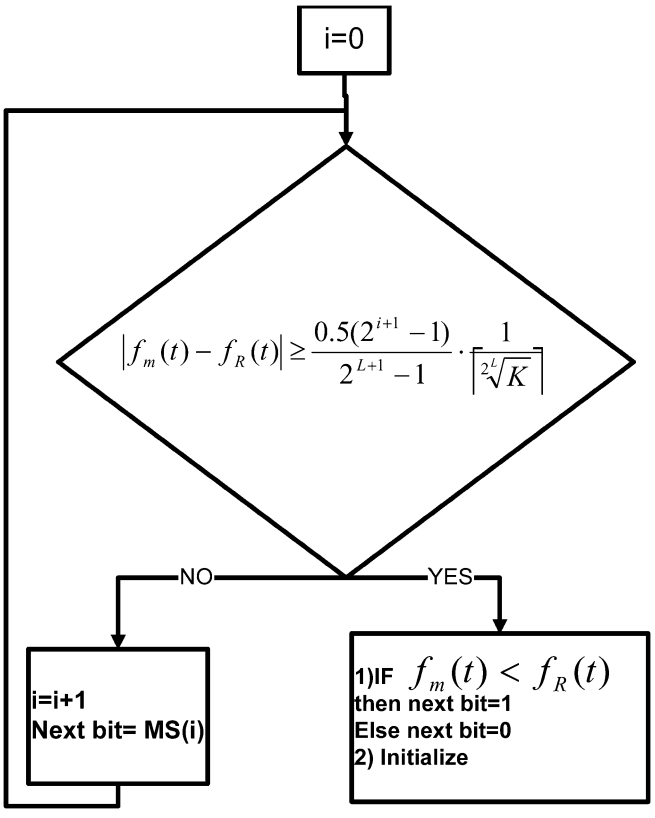

Fig. 11. The loss distributing algorithm.

Proof: Since Markov predictors of order L are a special case of machines with 2^L states, this is a trivial consequence of Theorem 4. □

APPENDIX II
PROOF OF LEMMA 5

Since, according to Lemma 4, each up-step starts from a bounded interval, we can bound the length of each of these up-steps $L_{US}(x)$ by

$$\begin{aligned}(1-x+\Delta(x))K^{-2/3} &\le L_{US}(x)\\ &\le \left(\frac{1-x+\Delta x}{1-K^{-2/3}}\right)K^{-2/3}+2\Delta(x)\end{aligned} \tag{41}$$

where $\Delta(x)$ is half the state gap and is proportional to $\frac{(x(1-x))^{1/2+\epsilon}}{K}$ by construction. The right-hand side term corresponds to up-steps that originate from x_{low} and the left-hand side terms correspond to up-steps that originate from x_{high}. By using series expansion and neglecting nondominant terms we get

$$\begin{aligned}(1-x)K^{-2/3} &\le L_{US}(x)\\ &\le (1-x)(K^{-2/3}+K^{-4/3})+O(\Delta(x)).\end{aligned} \tag{42}$$

In a similar fashion, one can bound the length of the down-steps (see (43) at the bottom of the page). Since each state gap has been traversed an equal amount of times on the way up and on the way down, the ratio between the number of up-steps and down-steps can be converted to a ratio of the length of the steps. Thus,

$$\xi_{UD}(x)=\frac{L_{DS}(x)}{L_{DS}(x)+L_{US}(x)}$$

which, by using (42), (43), and series expansion can be bounded by

$$A\le \xi_{UD}(x)\le B$$

where

$$B=\frac{xK^{-2/3}+O(xK^{-4/3})}{xK^{-2/3}+O(xK^{-4/3})+(1-x)K^{-2/3}} \tag{44}$$

$$A=\frac{xK^{-2/3}-O(\Delta(x))}{xK^{-2/3}+(1-x)K^{-2/3}+O(\Delta(x))+(1-x)K^{-4/3}} \tag{45}$$

and thus (multiplying denominator and numerator by $K^{2/3}$)

$$\begin{aligned}x-O(\Delta(x))K^{2/3}-xO\left(K^{-2/3}\right) &\le \xi_{UD}(x)\\ &\le x+xO\left(K^{-2/3}\right)\end{aligned}$$

which proves the lemma. □

APPENDIX III
A RECURSIVE FORMULA FOR THE VARIANCE OF THE ISW PROBABILITY ASSIGNMENT

We prove (31)

$$V_{n+1}=V_n\cdot\left(1-\frac{2}{K}\right)+\frac{E_n-E_n^2}{K^2}.$$

In the proof, we assume $x_n=1$ although the same can be proved for $x_n=0$.

Proof: We first show that the second moment obeys the recursive function

$$\begin{aligned}M_{n+1}&=\sum_{i=0}^{K}\Pr\left(p_{n+1}=\frac{i}{K}\right)\cdot\left(\frac{i}{K}\right)^2\\ &=\sum_{i=0}^{K}\left[\Pr\left(p_n=\frac{i}{K}\right)\cdot\frac{i}{K}\right.\\ &\qquad\left.+\Pr\left(p_n=\frac{i-1}{K}\right)\cdot\left(1-\frac{i-1}{K}\right)\right]\cdot\left(\frac{i}{K}\right)^2\\ &=\sum_{i=0}^{K}\left[\Pr\left(p_n=\frac{i}{K}\right)\cdot\frac{i}{K}\right]\cdot\left(\frac{i}{K}\right)^2\\ &\quad+\sum_{i=0}^{K}\left[\Pr\left(p_n=\frac{i}{K}\right)\cdot\left(1-\frac{i}{K}\right)\right]\cdot\left(\frac{i+1}{K}\right)^2\\ &=\sum_{i=0}^{K}\left[\Pr\left(p_n=\frac{i}{K}\right)\cdot\frac{i}{K}\right]\cdot\left(\frac{i}{K}\right)^2\\ &\quad+\sum_{i=0}^{K}\left[\Pr\left(p_n=\frac{i}{K}\right)\cdot\left(1-\frac{i}{K}\right)\right]\left(\frac{i^2+2i+1}{K^2}\right)\\ &=\sum_{i=0}^{K}\left[\Pr\left(p_n=\frac{i}{K}\right)\cdot\frac{i}{K}\right]\cdot\left(\frac{i}{K}\right)^2\\ &\quad+\sum_{i=0}^{K}\left[\Pr\left(p_n=\frac{i}{K}\right)\cdot\left(1-\frac{i}{K}\right)\right]\cdot\left(\frac{i^2}{K^2}\right)\\ &\quad+\sum_{i=0}^{K}\left[\Pr\left(p_n=\frac{i}{K}\right)\cdot\left(1-\frac{i}{K}\right)\right]\cdot\left(\frac{2i+1}{K^2}\right)\\ &=M_n+\frac{1}{K}\left(2E_n+\frac{1}{K}-2M_n-\frac{1}{K}E_n\right).\end{aligned}$$

Using the recursive function for the expectation (30)

$$(E_{n+1})^2=\left(E_n\cdot\left(1-\frac{1}{K}\right)+x_n\cdot\frac{1}{K}\right)^2$$

we get for the variance

$$\begin{aligned}V_{n+1}&=M_{n+1}-E_{n+1}^2\\ &=M_n+\frac{1}{K}\left(2E_n+\frac{1}{K}-2M_n-\frac{1}{K}E_n\right)\\ &\quad-\left(E_n\cdot\left(1-\frac{1}{K}\right)+\cdot\frac{1}{K}\right)^2.\end{aligned}$$

$$\begin{aligned}(x+\Delta(x))K^{-2/3}-O(\Delta(x))\le L_{DS}(x)&\le\frac{x-\Delta x}{1-K^{-2/3}}K^{-2/3}\\ &\Downarrow\\ xK^{-2/3}-O(\Delta(x))\le L_{DS}(x)&\le xK^{-2/3}+O(xK^{-4/3}).\end{aligned} \tag{43}$$

Rearranging terms

$$V_{n+1} = (M_n - E_n^2)\left(1 - \frac{2}{K}\right) + \frac{E_n - E_n^2}{K^2}$$
$$= V_n \cdot \left(1 - \frac{2}{K}\right) + \frac{E_n - E_n^2}{K^2}. \qquad \square$$

REFERENCES

[1] T. M. Cover, "Behavior of sequential predictors of binary sequences," in *Trans. 4th Prague Conf. Information Theory Statistical Decision Functions, Random Processes*, 1965, pp. 263–272.

[2] M. Datar, A. Gionis, P. Indyk, and R. Motwani, "Maintaining stream statistics over sliding windows (extended abstract)," in *Proc. 13th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA'02)*, 2002, pp. 635–644.

[3] L. D. Davisson, "Universal noiseless coding," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 783–795, Nov. 1973.

[4] M. Feder, "Gambling using a finite-state machine," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1459–1465, Sept. 1991.

[5] M. Feder and E. Federovski, "Prediction of binary sequences using finite memory," in *Proc. 1998 Int. Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 137.

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

[7] E. Federovski, "Branch prediction based on universal data compression algorithms," Master's thesis, Dept. Elec. Eng.–Syst., Tel-Aviv Univ., Tel-Aviv, Israel, 1998.

[8] J. Hannan, "Approximation to Bayes risk in repeated play," *Contributions to the Theory of Games*, vol. 3, pp. 97–139, 1957.

[9] M. E. Hellman and T. M. Cover, "Learning with finite memory," *Ann. Math. Statist.*, vol. 41, no. 3, pp. 765–782, 1970.

[10] J. C. Kieffer, "A unified approach to weak universal source coding," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 674–682, Nov. 1978.

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

[12] T. Leighton and R. L. Rivest, "Estimating a probability using finite memory," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 733–742, Nov. 1986.

[13] G. Longo and G. Galasso, "An application of informational divergence to Huffman codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 36–43, Jan. 1982.

[14] N. Merhav and M. Feder, "Universal prediction," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2124–2147, Oct. 1998.

[15] E. Meron, "Universal finite memroy prediction and coding of individual sequences," Master's thesis, Dept. Elec. Eng.–Syst., Tel-Aviv Univ., Tel-Aviv, Israel, 2003.

[16] D. Rajwan, "Universal finite memory coding of binary sequences," Master's thesis, Dept. Elec. Eng.–Syst., Tel-Aviv Univ., Tel-Aviv, Israel, 2000.

[17] D. Rajwan and M. Feder, "Universal finite memory machines for coding binary sequences," in *Proc. 2000 Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 113–122.

[18] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inform. Theory 30*, vol. IT-30, pp. 629–636, July 1984.

[19] J. Rissanen and G. G. Langdon, "Universal modeling and coding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 12–23, Jan. 1981.

[20] B. Y. Ryabko, "Imaginary sliding window as a tool for data compession," *Probl. Inform. Transm.*, pp. 156–163, Jan. 1996.

[21] Y. M. Shtar'kov, "Universal sequential coding of single messages," *Probl. Inform. Transm.*, vol. 23, no. 3, pp. 175–186, July–Sept. 1987.

[22] J. S. Vitter, "Optimal prefetching via data compression," *Proc. Foundations of Computer Science*, pp. 121–130, 1991.

[23] 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.