Discrete Universal Filtering Through Incremental Parsing

Erik Ordentlich[*], Tsachy Weissman[†], Marcelo J. Weinberger[*], Anelia Somekh–Baruch[‡], Neri Merhav[‡]

Abstract

In the discrete filtering problem, a data sequence over a finite alphabet is assumed to be corrupted by a discrete memoryless channel. The goal is to reconstruct the clean sequence, with as high a fidelity as possible, by way of causal processing of the noisy sequence alone, with the reconstruction at time t depending only on noisy observations occurring no later than t. We study a universal version of this problem in which no assumptions are made about the distribution of the clean data, which may even be non–stochastic. Using techniques from universal data compression, in particular, the incremental parsing rule of LZ78, we derive practical and efficient algorithms for the universal filtering of discrete sources. A *finite–memory* filter of order k has the property that the reconstruction at any time t is a time–invariant function only of noisy observations occurring between times $t-k$ and t, inclusive. We show that our universal filtering algorithms perform essentially as well, in an expected sense (with respect to the noise process), as the best finite–memory filter of any fixed order, determined with full knowledge of the actual clean data sequence, for all such data sequences. We also consider more general *finite–state* filters and show that any such filter is arbitrarily well approximated by a finite–memory filter of growing order, thereby establishing the universality of the proposed algorithms with respect to this larger class. This result can be viewed as the filtering analogue of the well known optimality of LZ78 relative to the class of finite–state compressors.

1 Introduction

There have been several successful applications of tools and techniques from data compression to other problem areas. A prominent instance of such cross–pollination is the use of the well known LZ78 incremental parsing rule [12] as the basis for practical and efficient algorithms for such diverse applications as gambling [2], prediction [3], and prefetching for memory caches [7]. These applications can be characterized as sequential decision problems, in which, at each time t, an action must be taken based on past observations of some data sequence, and the resulting action is evaluated against a new instance of the data sequence according to some loss function. The overall performance is measured by summing the losses incurred for each action–data

[*]HP Laboratories, 1501 Page Mill Rd.,Palo Alto, CA 94304, (eord,marcelo)@hpl.hp.com

[†]Department of Electrical Engineering, Stanford University, Stanford, CA, 94305, tsachy@stanford.edu

[‡]Department of Electrical Engineering, Technion, Israel–Institute of Technology, Haifa 32000, Israel, (anelia@techunix,merhav@ee).technion.ac.il

instance pair. The LZ78 incremental parsing rule provides an elegant algorithmic framework for collecting and processing counts of symbol occurrences in slowly growing contexts of previously occurring symbols. The resulting algorithms are universal in a sense analogous to that in which the basic LZ78 algorithm is universal in data compression. Specifically, in each application, the notion of a finite–memory machine of order k is defined.[1] Such a machine is constrained to base its action at time t on a time–invariant (randomized) function of data that occurred at times $t-k$ through $t-1$. The LZ78–based algorithms in the above applications then achieve essentially the same performance as the best finite–memory machine of any fixed order, matched to the specific data sequence, for every data sequence. In most cases, following [3], the notion of universality is strengthened by showing that finite–memory machines are no less powerful than more general finite–state machines driven by the data sequence.

In this work, we apply the LZ78–based parsing rule to the filtering problem, which can be formulated as follows. A data sequence $\{x_t\}$, with $x_t \in \mathcal{X}$ and $|\mathcal{X}|$ finite, is corrupted by a discrete memoryless channel (DMC) into a noisy data sequence $\{z_t\}$, $z_t \in \mathcal{Z}$, with $|\mathcal{Z}|$ finite. A filter estimates x_t by applying a (possibly randomized) function f_t to $z_1, \ldots, z_t$. Denoting the estimate by $\hat{x}_t = f_t(z_1, \ldots, z_t) \in \hat{\mathcal{X}}$, its accuracy is judged by evaluating a loss function, $\Lambda : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}^+$ on the pair $x_t, \hat{x}_t$. The overall performance of a filtering algorithm is assessed by summing the symbol-wise losses.

In analogy to the above sequential decision problems, the class of finite–state filters and the subclass of finite–memory filters can be defined. A key difference with respect to the above cases, however, is that here the machines operate on the noisy data, and not on the data they are actually trying to estimate. The notion of universality with respect to these classes of filters must be adapted as well, to account for the randomness in the noise. A filtering algorithm will be said to be universal with respect to a class of finite–state filters, if its expected loss, averaged over the noise process, is asymptotically no greater than the expected loss of the best finite–state filter of any fixed size in the class, matched to the specific underlying clean data sequence, for all such individual data sequences. We shall refer to this framework for assessing the performance of an algorithm as the *semi–stochastic setting* since it treats the noise as a random process, over which the loss is averaged, while it regards the clean data sequence as a non–random individual sequence.

Two obstacles stand in the way of obtaining a filtering algorithm that is universal in the above sense. The first, which also applies to the noise–free sequential decision problems mentioned above, is that the universal filter must generate its estimates causally, while the best finite–state filter is optimized non–causally, based on the entire data sequence. The second obstacle, which is unique to the noisy–setting, is that the universal algorithm must operate on the noisy data exclusively, while the best finite–state filter is optimized with full knowledge of the underlying clean sequence. Thus, any information about the clean sequence which is fundamental to the finite–state filter optimization must ultimately be "learned" from processing noisy observations. In particular, this precludes a direct application of the universal algorithms devised for the noise–free class of sequential decision problems to the filtering problem, since all of these algorithms, in deciding what action to take at time t, rely heavily on

[1] In previous papers (cf. eg. [3]) the adjective "Markov" is used instead of "finite–memory".

the ability to evaluate the loss of varying order finite–memory machines on portions of the data observed prior to time t. This reliance is not directly possible in the noisy setting, since the loss is a function of both the observable noisy data and the unobservable clean data.

We present two LZ78–motivated filtering algorithms, parameterized by DMC transition probabilities, that overcome the above obstacles and are shown to be universal with respect to the class of finite–state filters, in the sense described above, when their DMC parameters coincide with the actual DMC generating the noisy sequence. The algorithms and analysis apply to sources over arbitrary finite alphabets, arbitrary DMCs (subject to a full–rank assumption detailed below), and arbitrary loss functions. The first universal filtering algorithm we propose, which we shall call the dynamic IP universal filter (IP for incremental parsing), uses the data dependent LZ78 dynamic parsing rule to accumulate context dependent weights. This algorithm uses artificial randomization in mapping accumulated weights to filtering decisions.[2] The second algorithm, the static IP universal filter, uses a deterministic parsing derived from the LZ78 parsing of the "counting sequence", and does not require artificial randomization. A key component of our analysis is a mapping between filtering and prediction that allows us to "transfer" results from prediction theory to the filtering setting. For example, the mapping readily establishes the finite–memory approximability of any general finite–state filter.

A preliminary version of the basic approach pursued in this work appears in [1], which treats the special case of a binary source corrupted by a binary symmetric channel (BSC) with performance measured by the Hamming loss. The setting of causal filtering of noisy individual sequences was considered also in [10, Section 5], where a scheme was devised to compete with an arbitrary finite set of delay, memory, and rate constrained schemes (particularizing to the filtering problem when the rate and delay constraints are removed). Another related work is the recent [11] which treats the problem of universal denoising. This is similar to the filtering problem, except that a denoiser is allowed to base its estimate of the clean data symbol at time t on the entire noisy sequence, and, thus, unlike a filter, is not constrained to operate in a causal fashion. The problem of universal filtering with respect to the class of 0–th order finite–memory filters has a long history, and was originally referred to in the literature as the sequential compound Bayes problem (see [6] and references therein). Indeed, our static IP universal filter determines the filtering decision to apply in each context, by applying the algorithm of [6] to a subsequence of the noisy data. Finally, the problem of universal noisy prediction [8, 9] is also relevant. In the noisy prediction problem it is assumed that z_t is not available for estimating x_t so that the estimate of x_t is based only on $z_1, \ldots, z_{t-1}$. In [8] it is shown that the LZ78 driven universal prediction algorithm of [3] is, without modification, also universal for binary noisy prediction with respect to the class of finite–state noisy predictors. Such a result does not hold in the filtering setting, even if the statistics used to formulate the prediction of x_t in the algorithm of [3] include the sample z_t.

The paper is organized as follows. In Section 2 we introduce some basic notation. Section 3 formally defines the universal filtering problem and introduces some concepts that will be used later. Section 4 presents a correspondence between filtering

[2] We conjecture that universality is retained without the randomization.

and prediction that underlies many of our proofs, including the result of Section 5 showing that general finite–state filters are no more powerful, asymptotically, than finite–memory filters. Section 6 presents the dynamic IP universal filter, while Section 7 presents the non-randomized static IP universal filter. Some proofs are omitted in this extended abstract.

2 Notation and preliminaries

We will let $\mathcal{X}, \mathcal{Z}, \hat{\mathcal{X}}$ denote, respectively, the finite alphabets of the clean, noisy, and reconstructed source. We assume some given (arbitrary) total ordering on the elements of the alphabets, as well as on the elements of other finite sets that will be considered. The notation x^n will denote the sequence $x_1, \ldots, x_n$. A $|\mathcal{X}| \times |\hat{\mathcal{X}}|$ matrix Λ will denote the loss function, with the $x, \hat{x}$–th entry $\Lambda(x, \hat{x})$ specifying the loss incurred when estimating clean symbol x by $\hat{x}$. DMC channel transition probabilities will be denoted by a $|\mathcal{X}| \times |\mathcal{Z}|$ matrix Π, with the x, z–th entry $\Pi(x, z)$ specifying the probability that the channel output is z given that the input is x. The z–th column of Π will be denoted by π_z and the $\hat{x}$–th column of Λ by $\lambda_{\hat{x}}$. As in [11], Π is assumed to be of full row rank. For any finite set, $\mathcal{U}$, $\mathbf{R}^{\mathcal{U}}$ will denote the space of $|\mathcal{U}|$-dimensional column vectors with real-valued components indexed by the elements of $\mathcal{U}$ according to the total ordering, while $\mathcal{M}(\mathcal{U})$ will denote the simplex consisting of the elements of $\mathbf{R}^{\mathcal{U}}$ with non-negative components summing up to 1. The empirical distribution of a sequence $u^n \in \mathcal{U}^n$ will be denoted by $p_{u^n} \in \mathcal{M}(\mathcal{U})$, i.e., $p_{u^n}[u]$ is the fraction of appearances of u in u^n.

The following concepts and definitions are central to the description and analysis of the universal filtering algorithms in the sequel. Let h be a (column) vector–valued function $h : \mathcal{Z} \to \mathbf{R}^{\mathcal{X}}$ having the property that, for $a, b \in \mathcal{X}$,

$$\sum_{z \in \mathcal{Z}} h(z)[b]\Pi(a, z) = \delta(a, b) \stackrel{\triangle}{=} \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $h(z)[b]$ denotes the b–th component of $h(z)$ [6]. It is readily verified that our assumption of a channel matrix with full row rank guarantees the existence of a mapping h satisfying (1).[3] For any function v, $v : \mathcal{Z} \to \hat{\mathcal{X}}$, let $\rho(v) \in \mathbf{R}^{\mathcal{X}}$ denote the column vector with x–th component

$$\rho(v)[x] = \sum_z \Lambda(x, v(z))\Pi(x, z). \tag{2}$$

In words, $\rho(v)[x]$ is the expected loss of the estimator v when the underlying symbol is x. Let $\mathcal{V}$ denote the (finite) set of mappings taking $\mathcal{Z}$ into $\hat{\mathcal{X}}$, i.e., $\mathcal{V} = \left\{v : \mathcal{Z} \to \hat{\mathcal{X}}\right\}$ and, for $\xi \in \mathbf{R}^{\mathcal{Z}}$, let $B_h(\xi, \cdot) \in \mathcal{V}$ be defined by

$$B_h(\xi, \cdot) = \arg\min_{v \in \mathcal{V}} \left(\left[\sum_z \xi[z] h(z) \right]^T \cdot \rho(v) \right). \tag{3}$$

[3] In particular, for the case $|\mathcal{X}| = |\mathcal{Z}|$, there exists a unique such h given by $h(z)[b] = \Pi^{-1}(z, b)$.

Under the interpretation (which loosely applies in the sequel) that $\sum_z \xi[z]h(z)$ is a probability distribution on $\mathcal{X}$, $B_h(\xi, \cdot)$ corresponds to that mapping or estimate of x, as a function of z, that minimizes the expected loss, if the DMC Π inputs are distributed according to $\sum_z \xi[z]h(z)$.

3 The universal filtering problem

An individual data sequence x^n is assumed to be corrupted by a DMC with probability transition matrix Π resulting in a random noisy sequence Z^n. Thus, fixing x^n,

$$Pr(Z^n = z^n) = \prod_{t=1}^{n} \Pi(x_t, z_t). \tag{4}$$

A *filter* is a sequence of probability distributions $\hat{\mathbf{X}} = \{\hat{X}_t\}$, where $\hat{X}_t : \mathcal{Z}^t \to \mathcal{M}(\hat{\mathcal{X}})$. The interpretation is that, upon observing z^t, the reconstruction for the underlying, unobserved, x_t is given by the symbol $\hat{x}$ with probability $\hat{X}_t(z^t)[\hat{x}]$. A deterministic filter is a special case for which $\hat{X}_t(z^t)$ puts all the mass on a single (z^t dependent) reconstruction value. The *normalized cumulative loss* of the scheme $\hat{\mathbf{X}}$ on the individual pair (x^n, z^n) is defined by

$$L_{\hat{\mathbf{X}}}(x^n, z^n) = \frac{1}{n} \sum_{t=1}^{n} \sum_{\hat{x} \in \hat{\mathcal{X}}} \Lambda(x_t, \hat{x}) \hat{X}_t(z^t)[\hat{x}] \tag{5}$$

(each inner summation can be interpreted as an expectation w.r.t. the randomization). The expected loss of $\hat{\mathbf{X}}$ in a semi–stochastic setting on the individual clean data sequence x^n is defined by $EL_{\hat{\mathbf{X}}}(x^n, Z^n)$, with Z^n distributed according to (4).

We now formally define the notion of a finite–state filter: $\hat{\mathbf{X}}$ will be said to be a finite–state filter with finite state–space $\mathcal{S}$ if there exists a next–state function $g : \mathcal{S} \times \mathcal{Z} \to \mathcal{S}$, a reconstruction function $f : \mathcal{S} \times \mathcal{Z} \to \mathcal{M}$, and an initial state $s \in \mathcal{S}$ such that

$$\hat{X}_t(z^t) = f(s_t, z_t), \quad s_{t+1} = g(s_t, z_t), \quad s_1 = s. \tag{6}$$

We let $\mathcal{G}_{\mathcal{S}}$ denote the class of all finite–state filters with state space $\mathcal{S}$ and define

$$\phi_{\mathcal{S}}(x^n) = \min_{\hat{\mathbf{X}} \in \mathcal{G}_{\mathcal{S}}} EL_{\hat{\mathbf{X}}}(x^n, Z^n), \tag{7}$$

with the expectation on the right–hand side assuming the semi–stochastic setting. The quantity $\phi_{\mathcal{S}}(x^n)$ is thus the loss incurred in the semi–stochastic setting by the best finite–state filter on $\mathcal{S}$ for x^n. A *Markov filter of order k* is a finite–state filter with state space $\mathcal{S} = \mathcal{Z}^k$ and $s_t = z_{t-k}^{t-1}$. We let $\mathcal{F}_k$ denote the class of all finite–memory filters of order k and let $\mu_k(x^n)$ denote the loss incurred in the semi–stochastic setting by the best k–th order finite–memory filter for x^n, in analogy to (7). It can be shown that $\phi_{\mathcal{S}}(x^n)$ and $\mu_k(x^n)$ are achieved by deterministic filters.

The universal filtering problem is to construct a filtering algorithm $\hat{\mathbf{X}}$ satisfying

$$\limsup_{n \to \infty} [EL_{\hat{\mathbf{X}}}(x^n, Z^n) - \phi_{\mathcal{S}}(x^n)] \leq 0 \tag{8}$$

for all finite $\mathcal{S}$ and $x^\infty \in \mathcal{X}^\infty$, or, in words, to find an $\hat{\mathbf{X}}$ with expected loss in the semi–stochastic setting that is asymptotically no larger than $\phi_{\mathcal{S}}(x^n)$, for any fixed $\mathcal{S}$, and for all individual clean data sequences $x^\infty \in \mathcal{X}^\infty$.

4 Filtering as a prediction problem

In this section, we present a mapping from noise–free predictors to filters that will be useful in transferring known results from the noise–free prediction setting to the filtering setting. Let the finite sets $\mathcal{Y}$, $\mathcal{A}$ be, respectively, a source alphabet and a prediction alphabet (also referred to as the "action space"). A predictor F is a sequence of functions $F_t : \mathcal{Y}^{t-1} \to \mathcal{M}(\mathcal{A})$ with the interpretation that the prediction of the predictor F for time t is given by $a \in \mathcal{A}$ with probability $F_t(y^{t-1})[a]$. Assuming a given loss function $\ell : \mathcal{Y} \times \mathcal{A} \to \mathbf{R}$, for any $n > 0$ and $y^n \in \mathcal{Y}^n$, we define the *normalized cumulative loss* of the predictor F by

$$L_F(y^n) = \frac{1}{n} \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \ell(y_t, a) F_t(y^{t-1})[a]. \tag{9}$$

Consider predictors for the source alphabet $\mathcal{Y} = \mathcal{Z}$ and the action alphabet $\mathcal{A} = \mathcal{V}$, so that an action corresponds to selecting (at random) a function that maps $\mathcal{Z}$ to $\hat{\mathcal{X}}$. With any such predictor F we associate a filter $\hat{\mathbf{X}}^F$ in the following way:

$$\hat{X}_t^F(z^t)[\hat{x}] = \sum_{v : v(z_t) = \hat{x}} F_t(z^{t-1})[v]. \tag{10}$$

This association yields the following lemma, where we recall the existence of a vector valued function $h(z)$ satisfying (1).

Lemma 4.1 *Assume the semi-stochastic setting. For all $n > 0$ and $x^n \in \mathcal{X}^n$ we have*

$$EL_{\hat{\mathbf{X}}^F}(x^n, Z^n) = EL_F(Z^n), \tag{11}$$

where $L_F(z^n)$ is the normalized cumulative loss of an arbitrary predictor F (as defined in (9)) for the prediction problem with $\mathcal{Y} = \mathcal{Z}$, $\mathcal{A} = \mathcal{V}$, and the loss function

$$\ell(z, v) = h(z)^T \cdot \rho(v). \tag{12}$$

In words, Lemma 4.1 states that the *observable* $L_F(Z^n)$ is an unbiased estimate of $EL_{\hat{\mathbf{X}}^F}(x^n, Z^n)$ which is a function of x^n and therefore not observable.

5 Finite–memory approximations of finite–state filters

The correspondence between filtering and prediction provided by Lemma 4.1 allows us to easily establish Lemma 5.1 below, showing that the performance of the best finite–state filter can be approached by a finite–memory filter of growing order. From

Lemma 5.1 we conclude that if a filter is universal with respect to the class of finite–memory filters ($\phi_{\mathcal{S}}(x^n)$ replaced by $\mu_k(x^n)$, for any nonnegative integer k, in (8)), it is then also universal with respect to the larger class of finite–state filters.

The proof of Lemma 5.1 (and of Theorem 6.1 below) relies on known properties of finite–state and finite–memory predictors, which we now define. Using the notation of the previous section, F will be said to be a finite state predictor with state space $\mathcal{S}$ if there exists a next state function $\hat{g} : \mathcal{S} \times \mathcal{Y} \to \mathcal{S}$, an action function $\hat{f} : \mathcal{S} \to \mathcal{M}(\mathcal{A})$, and an initial state s such that

$$F_t(y^{t-1}) = \hat{f}(s_t), \quad s_{t+1} = \hat{g}(s_t, y_t), \quad s_1 = s. \tag{13}$$

A *Markov predictor of order* k is a special case for which $\mathcal{S} = \mathcal{Y}^k$ and $s_t = y_{t-k}^{t-1}$.

Lemma 5.1 *There exists a quantity C depending only on Λ and Π, such that for any nonnegative integers k and n, any finite state space $\mathcal{S}$, and any input sequence x^n,*

$$\mu_k(x^n) \le \phi_{\mathcal{S}}(x^n) + \left(\frac{C \log |\mathcal{S}|}{k+1} \right)^{1/2}. \tag{14}$$

Proof sketch: Let F^* be the deterministic finite–state predictor with state space $\mathcal{S}$, such that $\mathbf{X}^{F^*}$ (obtained via the association of Lemma 4.1) achieves $\phi_{\mathcal{S}}(x^n)$. Theorem 2 of [5] shows the existence of C depending only on ℓ, as defined by (12), such that for any k there exists a finite–memory predictor F_k satisfying $L_{F_k}(Z^n) \le L_{F^*}(Z^n) + ((C \log |\mathcal{S}|)/(k+1))^{1/2}$. The bound (14) follows by taking expectations and invoking Lemma 4.1 for the predictor–filter pairs (F^*, X^{F^*}) and (F_k, X^{F_k}). □

6 Dynamic IP universal filter

Like the LZ78 compression algorithm, the dynamic IP universal filter presented in this section parses the noisy observation sequence into distinct phrases such that each phrase is the shortest string which is not a previously parsed phrase. This procedure is most conveniently considered as a process of growing a tree, where each new phrase is represented by a leaf in the tree. Upon observing the first noisy symbol z_1, a tree is constructed consisting of a root and $|\mathcal{Z}|$ leaves (labeled by the symbols $z \in \mathcal{Z}$), with the leaf labeled by z_1 given weight 1, and the remaining leaves weight 0. At each step, the current tree is used to create an additional phrase by following the path corresponding to the incoming symbols. Once a leaf has been reached, the tree is extended at that point, making the leaf an internal node, adding its $|\mathcal{Z}|$ offspring to the tree, and giving the leaf corresponding to the observed symbol weight 1 while assigning weight 0 to the remaining leaves. The weight of each internal node is defined recursively as the sum of the weights of its $|\mathcal{Z}|$ offspring. The values of internal node weights can be updated efficiently by incrementing a counter associated with each node whenever that node is reached by the tree traversal determined by the incoming symbols. We let $w(z^t)$ denote the weight of the node reached by constructing and traversing the tree according to z^t, and $\bar{w}(z^{t-1}) \in \mathbf{R}^{\mathcal{Z}}$ denote the vector whose z–th

component is given by the weight of the child node of (the node associated with) z^{t-1} indexed by the symbol z. We define the dynamic IP universal filter $\hat{\mathbf{X}}^*$ by

$$\hat{X}_t^*(z^t)[\hat{x}] = \Pr\left\{B_h\left(\left[\bar{w}(z^{t-1}) + \mathbf{U}\sqrt{w(z^{t-1})+1}\right], z_t\right) = \hat{x}\right\}, \tag{15}$$

where $B_h(\cdot,\cdot)$ is defined by (3) and $\mathbf{U}$ is uniformly distributed on the cube $\left[0, \sqrt{\frac{6}{|\mathcal{Z}|}}\right]^{\mathcal{Z}}$.

Our main result is the following:

Theorem 6.1 *Assume the semi-stochastic setting. For all $x^\infty \in \mathcal{X}^\infty$, $n > 0$, $k \geq 0$,*

$$EL_{\hat{\mathbf{X}}^*}(x^n, Z^n) \leq \mu_k(x^n) + h_{max}\Lambda_{max}\left[\frac{k \cdot Ec(Z^n)}{n} + \sqrt{\frac{6|\mathcal{Z}|}{n}}E\sqrt{c(Z^n)}\right], \tag{16}$$

where $c(Z^n)$ denotes the number of phrases in the incremental parsing of Z^n, $\Lambda_{max} = \max_{x,\hat{x}}|\Lambda(x,\hat{x})|$, and $h_{max} = \max_z \|h(z)\|_1$. In particular, for all k,

$$\limsup_{n\to\infty}[EL_{\hat{\mathbf{X}}^*}(x^n, Z^n) - \mu_k(x^n)] \leq 0. \tag{17}$$

The proof of Theorem 6.1 is based on the filtering–prediction correspondence of Section 4 and also relies on known properties of a dynamic IP predictor P that operates as follows: P parses the source sequence, builds a tree, and assigns weights to the nodes exactly as does the dynamic IP universal filter detailed above. The prediction for time t is given by

$$P_t(y^{t-1})[a] = \Pr\left\{b\left(\bar{w}(y^{t-1}) + \mathbf{U}\sqrt{w(y^{t-1})+1}\right) = a\right\}, \tag{18}$$

where $b(\xi) \stackrel{\triangle}{=} \arg\min_{a\in\mathcal{A}}\sum_{y\in\mathcal{Y}}\xi[y]\ell(y,a)$ (also known as the Bayes response to ξ) and $\mathbf{U}$ is uniformly distributed on $\left[0, \frac{6}{|\mathcal{Y}|}\right]^{\mathcal{Y}}$. The predictor P is a slight variation on the predictor proposed in [7], which, in turn, builds on the predictors of [3, Section V] and [4]. It can be interpreted as an application of the prediction algorithm of [4] along dynamically determined subsequences obtained by classifying each time index t according to the node occupied by the tree traversal at time t.

Letting $\mathcal{M}_k$ denote the class of all finite–memory predictors of order k and $\ell_{max} = \max_{z,v}|\ell(z,v)|$, we have the following:

Lemma 6.2 *The dynamic IP predictor P defined in (18) satisfies, for all $y^\infty \in \mathcal{Y}^\infty$, $n > 0$, $k \geq 0$,*

$$L_P(y^n) \leq \min_{F\in\mathcal{M}_k} L_F(y^n) + \frac{k \cdot c(y^n) \cdot \ell_{\max}}{n} + \ell_{max}\sqrt{6|\mathcal{Y}|}\sqrt{\frac{c(y^n)}{n}}. \tag{19}$$

Lemma 6.2 is proved using the methodology pioneered in [3], and applied to this specific type of predictor in [7].

Proof of Theorem 6.1: Let P denote the incremental parsing predictor given by (18) with $\mathcal{Y} = \mathcal{Z}$, $\mathcal{A} = \mathcal{V}$ and the loss function $\ell(z, v) = h(z)^T \cdot \rho(v)$. By examining (3), (15), and (18) it is readily verified that $\hat{\mathbf{X}}^* = \hat{\mathbf{X}}^P$, where $\hat{\mathbf{X}}^P$ denotes the filter associated with the predictor P, as specified in (10). It is also readily verified that (under the association in (10)) to any k–th order finite–memory filter there exists a k–th order finite–memory predictor associated with it, and vice versa, thus, by applying Lemma 4.1,

$$E \min_{F \in \mathcal{M}_k} L_F(Z^n) \leq \min_{F \in \mathcal{M}_k} EL_F(Z^n) = \min_{F \in \mathcal{M}_k} EL_{\hat{\mathbf{X}}^F}(x^n, Z^n) = \min_{\hat{\mathbf{X}} \in \mathcal{F}_k} EL_{\hat{\mathbf{X}}}(x^n, Z^n) = \mu_k(x^n). \quad (20)$$

On the other hand, from Lemma 6.2 it follows that for all sample paths

$$L_P(Z^n) \leq \min_{F \in \mathcal{M}_k} L_F(Z^n) + \frac{k \cdot c(Z^n) \cdot \ell_{max}}{n} + \ell_{max} \sqrt{6|\mathcal{Z}|} \sqrt{\frac{c(Z^n)}{n}}. \quad (21)$$

The proof is completed by taking expectations over both sides of (21), invoking Lemma 4.1 for the left side, bounding the right side using (20), and noting that $\ell_{max} = \max_{z \in \mathcal{Z}, v \in \mathcal{V}} |h(z)^T \cdot \rho(v)| \leq h_{max} \Lambda_{max}$. Inequality (17) follows by the fact that $\max_{z^n} c(z^n) = O(n/\log n)$ [12]. □

7 Static IP universal filter

We now describe a filtering algorithm that attains the performance of the best k–th order finite–memory filter, for any k, without the use of artificial randomization. Our algorithm and analysis is based on the following fundamental result concerning the sequential compound Bayes problem.

Theorem 7.1 ([6]) *The deterministic filtering algorithm $\hat{\mathbf{X}}^0$ defined by*

$$\hat{X}_t^0(z^t)[\hat{x}] = \begin{cases} 1 & \text{if } \hat{x} = B_h(p_{z^t}, z_t) \\ 0 & \text{otherwise,} \end{cases}$$

satisfies $EL_{\hat{\mathbf{X}}^0}(x^n, Z^n) \leq \mu_0(x^n) + cn^{-1/2}$, for all x^n, where c is a constant depending only on Λ and h.

Thus, the algorithm $\hat{\mathbf{X}}^0$ competes effectively with the best 0–th order or memoryless filtering algorithm, determined with full knowledge of the clean sequence x^n.

We extend this algorithm to one which competes against the best k–th order finite–memory filter by applying the algorithm $\hat{\mathbf{X}}^0$ along subsequences of the noisy data z^n. Let $\mathcal{T}_{0,n}$ denote the set of indices corresponding to the starting points of phrases in an LZ78 incremental parsing of the "counting sequence" of length n, which consists of all distinct strings of length 1 over the alphabet $\mathcal{Z}$, followed by all distinct strings of length 2, and so on. Similarly, let $\mathcal{T}_{k,n}$ denote the set of indices for which the depth of the corresponding LZ78 tree traversal of the counting sequence is k. The static IP universal filtering algorithm $\hat{\mathbf{X}}^s$ partitions $\mathcal{T}_{k,n}$, for each k, into subsequences based

on the value of z_{t-k}^{t-1} for $t \in \mathcal{T}_{k,n}$, and applies the filtering algorithm $\hat{\mathbf{X}}^0$ to the symbols along each such subsequence (one subsequence for each k and aligned occurrence of each possible k–tuple in z^n), treating each subsequence independently.

Formally, let $t_1 = 1,\ t_{i+1} = t_i + i|\mathcal{Z}|^i$, and define

$$\begin{aligned} I(t) &\stackrel{\triangle}{=} \max\{i : t_i \le t\}, \\ K(t) &\stackrel{\triangle}{=} t - t_{I(t)} \mod I(t) \ \ (= \text{context length at } t), \\ \mathcal{T}_{k,n} &\stackrel{\triangle}{=} \{t \le n : K(t) = k\}, \qquad \mathcal{T}_{k,n,\tilde{z}^k}(z^n) \stackrel{\triangle}{=} \{t \in \mathcal{T}_{k,n} : z_{t-k}^{t-1} = \tilde{z}^k\}. \end{aligned}$$

We define the deterministic sequential filtering algorithm $\hat{\mathbf{X}}^s$ by

$$\hat{X}_t^s(z^t) = \hat{X}^0_{|\mathcal{T}_{K(t),t,z_{t-K(t)}^{t-1}}(z^t)|}(z[\mathcal{T}_{K(t),t,z_{t-K(t)}^{t-1}}(z^t)]), \tag{22}$$

where, for any set of distinct integers $A = \{i_1 < \ldots < i_m\}$, $z[A] \stackrel{\triangle}{=} (z_{i_1}, z_{i_2}, \ldots, z_{i_m})$.

One can interpret the algorithm $\hat{\mathbf{X}}^s$ as engaging in an LZ78-like incremental parsing, but where the parsing is deterministic and not data driven. The following result bounds the performance of the static IP universal filter $\hat{\mathbf{X}}^s$.

Theorem 7.2 *Assume the semi-stochastic setting. For all $x^\infty \in \mathcal{X}^\infty$ and any fixed k,*

$$EL_{\hat{\mathbf{X}}^s}(x^n, Z^n) \le \mu_k(x^n) + O((\log n)^{-1/2}). \tag{23}$$

References

[1] A. Baruch and N. Merhav. Universal filtering and prediction of individual sequences corrupted by noise. *Proceedings, 37–th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 22–24, 1999.

[2] M. Feder. Gambling using a finite state machine. *IEEE Trans. Inform. Theory*, 37(5):1459–1465, September 1991.

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

[4] J. F. Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, (3):97–139, 1957. Princeton University Press.

[5] N. Merhav and M. Feder. Universal schemes for sequential decisions from individual data sequences. *IEEE Trans. Inform. Theory*, 39:1280–1292, July 1993.

[6] J. Van Ryzin, The sequential compound decision problem with $m \times n$ finite loss matrix, *Ann. Math. Stat.*, vol. 37, 1966, pp. 954–975.

[7] M. J. Weinberger and E. Ordentlich, On–line decision making for a class of loss functions via Lempel–Ziv parsing, *Proceedings, Data Compression Conference*, March 28–30, 2000, Snowbird, Utah.

[8] T. Weissman and N. Merhav, Universal prediction of individual binary sequences in the presence of noise, *IEEE Trans. Inform. Theory*, 47(6):2151–2173, September 2001.

[9] T. Weissman, N. Merhav, and A. Baruch, Twofold universal prediction schemes for achieving the finite-state predictability of a noisy individual binary sequence, *IEEE Trans. Inform. Theory*, 47(5):1849–1866, July 2001.

[10] T. Weissman and N. Merhav. Finite-delay lossy coding and filtering of individual sequences, *IEEE Trans. Inform. Theory*, 48(3):721–733, March 2002.

[11] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, M. J. Weinberger, Universal discrete denoising: known channel, *HP Labs. Tech. Rep.*, HPL-2003-29, February 2003. Submitted to *IEEE Trans. Inform. Theory.*

[12] J. Ziv and A. Lempel, Compression of individual sequences via variable–rate coding, *IEEE Trans. Inform. Theory*, 24:530–536, September 1978.