

Efficient Pruning of Bi-Directional Context Trees with Applications to Universal Denoising and Compression

Erik Ordentlich
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA
e-mail: eord@hpl.hp.com

Marcelo J. Weinberger
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA
e-mail: marcelo@hpl.hp.com

Tsachy Weissman*
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA
e-mail: tsachy@stanford.edu

Abstract — The classical framework of context-tree models, customary in sequential decision problems such as compression and prediction, is generalized to a setting in which the observations are multi-tracked or multi-directional, and for which it may be beneficial to consider contexts comprised of possibly differing numbers of symbols from each track or direction. The notion of a bi-directional context set is formalized and the generalization of the classical context-tree-based representation for a well defined set of bi-directional contexts is presented, together with an efficient dynamic programming algorithm for determining the best set of bi-directional contexts for a given individual sequence, maximum context depth, and loss function. After briefly describing how this framework can be applied to universal data compression, we focus on its application to universal denoising, where we pair the proposed framework with a new technique for estimating the loss of a denoising algorithm based only on noisy observations.

I. INTRODUCTION

The classical context modeling technique [1, 2] used in data compression and other sequential decision problems decomposes a data sequence into a set of subsequences based on the occurrence of certain substrings (sequences) of symbols, the substrings being elements of a context set. Subsequent processing steps such as probability assignment or estimation then treat the resulting subsequences independently. Let \mathcal{X} be the data sequence alphabet and let $\mathcal{S} \subset \mathcal{X}^*$ denote the finite set of finite length strings of symbols comprising the context set, where \mathcal{X}^* is the set of all finite length strings (including the empty string) over the alphabet \mathcal{X} . Let k be the length of the longest string in \mathcal{S} . For $\mathbf{s} \in \mathcal{S}$ define $\mathcal{P}(\mathbf{s}) = \{\mathbf{x} \in \mathcal{X}^k : \mathbf{x}^{|\mathbf{s}|} = \mathbf{s}\}$, where \mathbf{x}^l is the l symbol prefix of \mathbf{x} and $|\mathbf{s}|$ denotes the length of \mathbf{s} . A valid context set \mathcal{S} must satisfy the following two properties: (1) $\cup_{\mathbf{s} \in \mathcal{S}} \mathcal{P}(\mathbf{s}) = \mathcal{X}^k$ (exhaustive) and (2) $\mathcal{P}(\mathbf{s}) \cap \mathcal{P}(\mathbf{s}') = \emptyset$ for any pair $\mathbf{s} \neq \mathbf{s}'$ (dis-joint). Given a data sequence $\mathbf{x}^n = (x_1, x_2, \dots, x_n)$, the subsequence of symbols associated with a context \mathbf{s} in a well defined context set \mathcal{S} consists of those symbols x_i whose preceding symbols satisfy $\mathbf{s} = \mathbf{x}_{i-|\mathbf{s}|}^{i-1}$, where $\mathbf{x}_i^m = (x_i, x_{i+1}, \dots, x_m)$. For each $\mathbf{s} \in \mathcal{S}$ let $\mathbf{x}(\mathbf{s})$ denote the subsequence of data symbols associated in this manner with \mathbf{s} . The “exhaustive” and “dis-joint” properties guarantee that each x_i belongs to one and only one such subsequence. Any given $\mathbf{x}(\mathbf{s})$ may be empty however. It is well known that the “exhaustive” and “dis-joint” properties also imply that the set of strings in \mathcal{S} can

be represented as the leaves of a (context) tree having nodes $\mathbf{n} \in \mathcal{X}^*$ where each node \mathbf{n} is either a leaf or has the $|\mathcal{X}|$ children $\{\mathbf{n}x : x \in \mathcal{X}\}$. Context tree models are extensively studied in [2].

In many applications of context models the processing of a subsequence of data symbols results in a numerical loss. In compression, this loss might be the ideal code length corresponding to the probability assigned to the subsequence by a sequential probability assignment procedure, such as one based on the Krichevsky-Trofimov (KT) estimator [3]. In a prediction application, the loss might be the prediction error incurred by a sequential prediction algorithm, such as that of Hannan [4], applied to the subsequence. To each subsequence we associate a *weight*, given by the loss incurred by processing the subsequence. A useful computation that arises in such settings is the determination of a context set, subject to a constraint on the maximum context length, that, for a given individual data sequence \mathbf{x}^n , minimizes the sum of the weights of the resulting set of context dependent subsequences. More formally, assume that an application induces a weight function λ on sequences of symbols over \mathcal{X} .¹ Given λ , an individual data sequence \mathbf{x}^n , and a maximum context length, of interest is that valid context set $\mathcal{S} \subset \mathcal{X}^{k*}$ that minimizes $\sum_{\mathbf{s} \in \mathcal{S}} \lambda(\mathbf{x}(\mathbf{s}))$, where \mathcal{X}^{k*} is the set of strings over \mathcal{X} of length at most k . An efficient dynamic programming based algorithm that relies on the context tree representation of valid context sets is known for carrying out this computation [5].

In the general sequential decision setting, the \mathbf{x}^n that is the target of such a computation may be training data and the context set derived by the preceding computation might be a good candidate for use on actual data. A “plug-in” sequential decision approach (“plugging in” what is best for the past) employs training data consisting of previously observed data and the computation is repeated often as new data is observed. In the compression setting the computation of the best context set is central to several *two-part* universal source codes proposed in the literature (see [6] and references therein). One weight function on sequences arising in some of these cases consists of the ideal code length induced by the KT probability assignment (which is obtained by accumulation of instantaneous losses) plus a constant (subsequence independent) offset to account for the cost of describing the context set itself. The overall two part code then consists first of a description of the best choice of contexts \mathcal{S}_{opt} determined via the aforementioned computation, and second of the sequence compressed via arithmetic coding based on symbol probabilities generated by context dependent KT estimators.

In this work we generalize the above classical context mod-

*This author is also with Hewlett-Packard Laboratories, Palo Alto, CA 94304, USA.

¹The weight function is often obtained by accumulation of instantaneous losses corresponding to the symbols in the sequence.

eling framework to a setting in which the observations are multi-tracked, multi-sided, or multi-directional and for which it may be beneficial to consider contexts comprised of possibly differing numbers of symbols from each track or side. We formalize the notion of a bi-directional context set, present a generalization of the above tree-based representation for a well defined set of bi-directional contexts, and describe an analogue of the above algorithm for determining the best set of bi-directional contexts for a given individual sequence and subsequence weight function. After briefly describing how this framework can be applied to universal data compression, we present an in-depth application to universal denoising in which we pair the framework with a new technique for estimating the loss of a denoising algorithm based only on noisy observations.

II. BI-DIRECTIONAL CONTEXT SETS: DEFINITION AND STRUCTURE

For simplicity we present our framework assuming two directions, a “left” and a “right.” Treatment of the m -directional case, $m > 2$ is deferred to the full paper [7]. Formally, our setting involves a data sequence $\mathbf{x} = \{x_i : i \in \mathcal{I}\}$ where, for each i , left and right directional sequences $\mathbf{y}_\ell^{(i)}$ and $\mathbf{y}_r^{(i)}$ are available for forming contexts. In one example of such a setting, $\mathcal{I} = \{1, 2, 3, \dots\}$ and \mathbf{x} consists of two tracks so that $x_i = (x_{L,i}, x_{R,i})$ and the left and right directional sequences correspond to $\mathbf{y}_\ell^{(i)} = x_{L,i-1}, x_{L,i-2}, \dots$ and $\mathbf{y}_r^{(i)} = x_{R,i-1}, x_{R,i-2}, \dots$. In a digital image compression or processing setting, $\mathcal{I} = \{1, 2, 3, \dots\} \times \{1, 2, 3, \dots\}$, $x_{i,j}$ represents the pixel value in row i and column j of the image, and $\mathbf{y}_\ell^{(i,j)}$ may be set to $x_{i,j-1}, x_{i,j-2}, \dots$ (the sequence of pixel values appearing to the left in row i) while $\mathbf{y}_r^{(i,j)}$ may be set to $x_{i-1,j}, x_{i-2,j}, \dots$ (the sequence of pixel values appearing above in column j). The denoising setting described in more detail in Section IV involves $\mathcal{I} = \{1, 2, 3, \dots\}$ with $\mathbf{y}_\ell^{(i)} = x_{i-1}, x_{i-2}, \dots$ and $\mathbf{y}_r^{(i)} = x_{i+1}, x_{i+2}, \dots$.

Paralleling the classical case, a bi-directional context set $\mathcal{S} \subseteq \mathcal{X}^* \times \mathcal{X}^*$ is a finite set of ordered pairs of finite length strings over the observation alphabet specifying the bi-directional contexts. Let k be the length of the longest string in any pair in \mathcal{S} . For $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}$ define $\mathcal{P}(\mathbf{s}_\ell, \mathbf{s}_r) = \{(\mathbf{x}^k, \mathbf{y}^k) \in \mathcal{X}^k \times \mathcal{X}^k : \mathbf{x}^{|\mathbf{s}_\ell|} = \mathbf{s}_\ell, \mathbf{y}^{|\mathbf{s}_r|} = \mathbf{s}_r\}$, the pairs of strings of length k whose first and second components respectively have \mathbf{s}_ℓ and \mathbf{s}_r as prefixes.

For a bi-directional context set to be well defined or valid, the set \mathcal{S} must satisfy the following generalizations of the “exhaustive” and “disjoint” conditions from the uni-directional case: (1) $\cup_{(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}} \mathcal{P}(\mathbf{s}_\ell, \mathbf{s}_r) = \mathcal{X}^k \times \mathcal{X}^k$ (exhaustive) and (2) $\mathcal{P}(\mathbf{s}_\ell, \mathbf{s}_r) \cap \mathcal{P}(\mathbf{s}'_\ell, \mathbf{s}'_r) = \emptyset$ for any pair $(\mathbf{s}_\ell, \mathbf{s}_r) \neq (\mathbf{s}'_\ell, \mathbf{s}'_r)$ (disjoint).

Given a data sequence \mathbf{x} , the subset of symbols associated with a context pair $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}$ consists of those symbols x_i whose corresponding context formation sequences satisfy

$$\mathbf{s}_\ell = [\mathbf{y}_\ell^{(i)}]^{|\mathbf{s}_\ell|}$$

and

$$\mathbf{s}_r = [\mathbf{y}_r^{(i)}]^{|\mathbf{s}_r|}.$$

As in the classical uni-directional case, the new “exhaustive” and “disjoint” properties guarantee that each x_i belongs to one and only one such subset. For each $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}$, let

$\mathbf{x}(\mathbf{s}_\ell, \mathbf{s}_r)$ denote the subset of data symbols associated in the above manner with $(\mathbf{s}_\ell, \mathbf{s}_r)$.

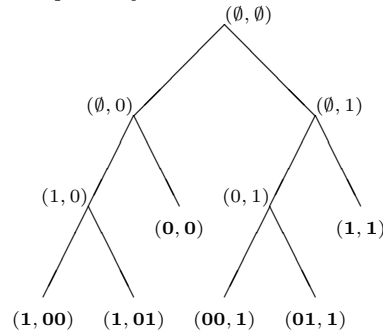
As in the uni-directional case, the new “exhaustive” and “disjoint” properties lead to a tree based representation of a valid bi-directional context set \mathcal{S} . In the bi-directional case the tree, which is defined in the next lemma and which we shall refer to as a bi-directional context tree, has a somewhat different structure and is not necessarily unique for a given \mathcal{S} .

Lemma II.1 *The string pairs in a valid bi-directional context set \mathcal{S} can be represented as the leaves of a rooted tree (bi-directional context tree) having nodes in $\mathcal{X}^* \times \mathcal{X}^*$ where the root node is the pair of empty strings (\emptyset, \emptyset) and each node $n = (\mathbf{s}_\ell, \mathbf{s}_r)$ is either a leaf or the set of its children is either $\{(\mathbf{s}_\ell, \mathbf{s}_r x) : x \in \mathcal{X}\}$ or $\{(\mathbf{s}_\ell x, \mathbf{s}_r) : x \in \mathcal{X}\}$.*

We note for future reference the easily seen fact that, conversely, the leaves of any bi-directional context tree, as defined in Lemma II.1, determine a valid bi-directional context set. Additionally, the structure of a bi-directional context tree can be inferred from its nodes. In the sequel a bi-directional context tree will be represented using the set of its nodes.

The following is an example of a valid bi-directional context set and an associated bi-directional context tree, as guaranteed by Lemma II.1. Note that the sets of strings formed from either the left components or the right components of the pairs in \mathcal{S} fail to constitute valid uni-directional context sets.

Example II.2 *Let $\mathcal{X} = \{0, 1\}$ and $\mathcal{S} = \{(0, 0), (1, 00), (1, 01), (00, 1), (01, 1), (1, 1)\}$. An associated bi-directional context tree is given by*



A bi-directional context tree can be interpreted as specifying, for a valid \mathcal{S} , a recursive sequence of splittings of the set $\mathcal{X}^k \times \mathcal{X}^k$ into the sets $\mathcal{P}(\mathbf{s}_\ell, \mathbf{s}_r)$ for $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}$. Lemma II.1 thus shows that valid bi-directional context sets can be obtained as a sequence of splittings, where the three possible splittings are “not splitting” and splitting based on the value of the next left context string symbol or the value of the next right context string symbol. In [8], weighting and pruning algorithms are proposed for classes of context sets that are generated according to a variety of splitting rules. While the splitting rule relevant to bi-directional context sets is not specifically considered in [8], it is straightforward to extend the algorithms in [8] for finding optimal context sets (and for weighting among these sets in compression applications) to this case. For completeness, in the next section we concretely describe such an algorithm.

III. BI-DIRECTIONAL PRUNING

Applications similar to those described for uni-directional context models motivate the following computation of the best bi-directional context set for a given individual sequence. Given a sequence \mathbf{x}^n , a set of context formation sequences $\{(\mathbf{y}_\ell^{(i)}, \mathbf{y}_r^{(i)})\}$, a subsequence weight function λ , and a maximum context length k , let

$$L(\mathcal{S}) = \sum_{(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}} \lambda(\mathbf{x}(\mathbf{s}_\ell, \mathbf{s}_r)).$$

Of interest is

$$\mathcal{S}_{\text{opt}} = \arg \min_{\text{valid } \mathcal{S} \subseteq \mathcal{X}^k \times \mathcal{X}^k} L(\mathcal{S}),$$

where ties are broken according to a deterministic but arbitrary rule. Below we present an efficient dynamic programming algorithm for computing \mathcal{S}_{opt} .

For any pair $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{X}^{k^*} \times \mathcal{X}^{k^*}$ we define the weight of $(\mathbf{s}_\ell, \mathbf{s}_r)$ as

$$w(\mathbf{s}_\ell, \mathbf{s}_r) = \lambda(\mathbf{x}(\mathbf{s}_\ell, \mathbf{s}_r)),$$

or the weight of the subsequence of data symbols whose left and right context formation sequences have prefixes equal to $(\mathbf{s}_\ell, \mathbf{s}_r)$. We set $w(\mathbf{s}_\ell, \mathbf{s}_r) = 0$ if this subsequence is empty. It is not difficult to see that for a valid context set \mathcal{S} , $L(\mathcal{S})$ is equal to the sum of the weights of the elements of \mathcal{S} and, correspondingly, of the leaves of a representative bi-directional context tree.

For any bi-directional context tree T , as defined in Lemma II.1, let $w(T)$ denote the weight of T defined as the sum of the weights of the leaves. Let \mathcal{T}_{opt} be the set of bi-directional context trees with nodes in $\mathcal{X}^{k^*} \times \mathcal{X}^{k^*}$ having minimal weight. In general, this set will have cardinality greater than one, even when \mathcal{S}_{opt} is unique, due to the multiple representations of the context set. Lemma II.1 and the above then imply that \mathcal{S}_{opt} corresponds to the leaves of an element of \mathcal{T}_{opt} , thereby reducing the problem of determining \mathcal{S}_{opt} to the problem of determining an element of \mathcal{T}_{opt} .

Given any pair $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{X}^{k^*} \times \mathcal{X}^{k^*}$ a bi-directional context subtree rooted at $(\mathbf{s}_\ell, \mathbf{s}_r)$ has nodes in $\{(\mathbf{s}_\ell \mathbf{s}'_\ell, \mathbf{s}_r \mathbf{s}'_r) : \mathbf{s}'_\ell \in \mathcal{X}^{(k-|\mathbf{s}_\ell|)^*}, \mathbf{s}'_r \in \mathcal{X}^{(k-|\mathbf{s}_r|)^*}\}$, where, as in the definition of a full bi-directional context tree, a node $(\tilde{\mathbf{s}}_\ell, \tilde{\mathbf{s}}_r)$ is either a leaf or the set of its children is either $\{(\tilde{\mathbf{s}}_\ell, \tilde{\mathbf{s}}_r x) : x \in \mathcal{X}\}$ or $\{(\tilde{\mathbf{s}}_\ell x, \tilde{\mathbf{s}}_r) : x \in \mathcal{X}\}$. Let $\mathcal{T}(\mathbf{s}_\ell, \mathbf{s}_r)$ denote the set of bi-directional subtrees rooted at $(\mathbf{s}_\ell, \mathbf{s}_r)$. Extend the definition of the weight function $w(T)$ to bi-directional subtrees T in the obvious way and let $\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r)$ be the subset of bi-directional subtrees in $\mathcal{T}(\mathbf{s}_\ell, \mathbf{s}_r)$ having minimal weight. We then have the following principle of optimality.

Lemma III.1 For any pair $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{X}^{k^*} \times \mathcal{X}^{k^*}$

$$\min_{T \in \mathcal{T}(\mathbf{s}_\ell, \mathbf{s}_r)} w(T) = \min \left[w(\mathbf{s}_\ell, \mathbf{s}_r), \sum_{x \in \mathcal{X}} \min_{T' \in \mathcal{T}(\mathbf{s}_\ell x, \mathbf{s}_r)} w(T'), \sum_{x \in \mathcal{X}} \min_{T' \in \mathcal{T}(\mathbf{s}_\ell, \mathbf{s}_r x)} w(T') \right], \quad (1)$$

where we take $\mathcal{T}(\tilde{\mathbf{s}}_\ell, \tilde{\mathbf{s}}_r)$ to be empty if $(\tilde{\mathbf{s}}_\ell, \tilde{\mathbf{s}}_r) \notin \mathcal{X}^{k^*} \times \mathcal{X}^{k^*}$ and the minimum of any function over an empty set to be infinity.

If the minimum on the right hand side of (1) is achieved by the first term, then the tree $\{(\mathbf{s}_\ell, \mathbf{s}_r)\} \in \mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r)$. Otherwise,

$$\{(\mathbf{s}_\ell, \mathbf{s}_r)\} \cup \bigcup_{x \in \mathcal{X}} T_x \in \mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r)$$

where, if the minimum is achieved by the second term, T_x denotes any member of $\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell x, \mathbf{s}_r)$, whereas if the minimum is achieved by the third term, T_x denotes any member of $\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r x)$.

Lemma III.1 parallels a similar principle of optimality for the uni-directional case (where the minimum in (1) is over two values), and suggests the following dynamic programming algorithm for determining an element of \mathcal{T}_{opt} . We shall refer to Algorithm III.2 as carrying out a bi-directional context tree pruning.

Algorithm III.2

```

for each  $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{X}^k \times \mathcal{X}^k$ 
  determine  $w(\mathbf{s}_\ell, \mathbf{s}_r)$ 
   $\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r) = \{(\mathbf{s}_\ell, \mathbf{s}_r)\}$ 
end
for  $m = 2k-1$  to 0
  for each  $(\mathbf{s}_\ell, \mathbf{s}_r)$  with  $|\mathbf{s}_\ell| + |\mathbf{s}_r| = m$ 
    determine  $w(\mathbf{s}_\ell, \mathbf{s}_r)$ 
     $N = w(\mathbf{s}_\ell, \mathbf{s}_r)$ 
     $R = \sum_{x \in \mathcal{X}} w(\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r x))$ 
     $L = \sum_{x \in \mathcal{X}} w(\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell x, \mathbf{s}_r))$ 
     $M = N$ ,  $\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r) = \{(\mathbf{s}_\ell, \mathbf{s}_r)\}$ 
    if  $R < M$  then
       $M = R$ 
       $\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r) = \{(\mathbf{s}_\ell, \mathbf{s}_r)\} \cup \bigcup_{x \in \mathcal{X}} \mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r x)$ 
    end
    if  $L < M$  then
       $M = L$ 
       $\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r) = \{(\mathbf{s}_\ell, \mathbf{s}_r)\} \cup \bigcup_{x \in \mathcal{X}} \mathcal{T}_{\text{opt}}(\mathbf{s}_\ell x, \mathbf{s}_r)$ 
    end
  end
end

```

In the algorithm, $\mathcal{T}_{\text{opt}}(\mathbf{s}_\ell, \mathbf{s}_r)$ is taken to be empty for $(\mathbf{s}_\ell, \mathbf{s}_r) \notin \mathcal{X}^{k^*} \times \mathcal{X}^{k^*}$ and the weight of an empty tree is taken to be infinity. The following theorem, which follows from Lemmas II.1 and III.1, establishes that Algorithm III.2 carries out the desired computation.

Theorem III.3 The tree $\mathcal{T}_{\text{opt}}(\emptyset, \emptyset)$ generated by Algorithm III.2 is an element of \mathcal{T}_{opt} and its leaves constitute \mathcal{S}_{opt} .

In many applications, such as two-part codes with KT estimation and the denoising application of Section IV, $w(\mathbf{s}_\ell, \mathbf{s}_r)$ is a relatively simple function of the vector of counts

$$\mathbf{m}(\mathbf{x}^n, \mathbf{s}_\ell, \mathbf{s}_r)[x] = \sum_{i: x_i \in \mathbf{x}(\mathbf{s}_\ell, \mathbf{s}_r)} 1(x_i = x),$$

for all $x \in \mathcal{X}$. In these cases, the sequence \mathbf{x}^n need only be processed to determine the counts of the contexts of maximal length, as done in the first **for** loop. The counts for the shorter contexts can then be determined as the sum of the counts

of either the left children or right children. Specifically, for $(\mathbf{s}_\ell, \mathbf{s}_r)$ with $|\mathbf{s}_\ell| + |\mathbf{s}_r| < 2k$,

$$\begin{aligned} \mathbf{m}(\mathbf{x}^n, \mathbf{s}_\ell, \mathbf{s}_r) &= \sum_{x' \in \mathcal{X}} \mathbf{m}(\mathbf{x}^n, \mathbf{s}_\ell, \mathbf{s}_r, x') \\ &= \sum_{x' \in \mathcal{X}} \mathbf{m}(\mathbf{x}^n, \mathbf{s}_\ell x', \mathbf{s}_r). \end{aligned}$$

Finally, we note that the complexity of Algorithm III.2 can be reduced by restricting processing only to those contexts that actually occur in the sequence.

IV. APPLICATION TO DENOISING

The bi-directional context tree pruning algorithm described in Section III can be applied to context-based denoising, in particular to enhance the DUDE algorithm proposed in [9]. In the (semi-stochastic) universal denoising setting, we consider an individual sequence \mathbf{x}^n , which is corrupted by a (probabilistic) memoryless channel with known transition probability matrix $\mathbf{\Pi}$. For simplicity, we will assume that the input and output alphabets coincide, so that $\mathbf{\Pi} = \{\Pi(x, z)\}_{x, z \in \mathcal{X}}$. It is also assumed that $\mathbf{\Pi}$ is invertible. A (noisy) sequence \mathbf{z}^n is observed at the output of the channel, and the goal is to denoise \mathbf{z}^n without knowledge of the (clean) sequence \mathbf{x}^n , to obtain a sequence $\hat{\mathbf{x}}^n \in \mathcal{X}^n$, where a given loss function $\Lambda : \mathcal{X}^2 \rightarrow [0, \infty)$, represented by the matrix $\mathbf{\Lambda} = \{\Lambda(x, z)\}_{x, z \in \mathcal{X}}$, determines the loss incurred by estimating each symbol x_i with the symbol \hat{x}_i , $1 \leq i \leq n$. The cumulative loss is determined by adding the instantaneous losses over time. The denoiser is allowed to observe the entire sequence \mathbf{z}^n before starting to make its decisions. A family of denoisers, parameterized by a nonnegative integer parameter k , is proposed in [9]. For a given k , the denoiser output $\hat{x}_i^*(z_i)$ at time i for a noisy input z_i , $k+1 \leq i \leq n-k$, is given by

$$\hat{x}_i^*(z_i) = \arg \min_{\hat{x} \in \mathcal{X}} \mathbf{m}(\mathbf{z}^n, z_{i-k}^{i-1}, z_{i+1}^{i+k}) \mathbf{\Pi}^{-1} [\boldsymbol{\lambda}_{\hat{x}} \odot \boldsymbol{\pi}_{z_i}] \quad (2)$$

where $\mathbf{m}(\mathbf{z}^n, z_{i-k}^{i-1}, z_{i+1}^{i+k})$ is a $|\mathcal{X}|$ -dimensional row vector whose β -th component, $\beta \in \mathcal{X}$, is the number of appearances of the string $z_{i-k}^{i-1} \beta z_{i+1}^{i+k}$ in \mathbf{z}^n , $\boldsymbol{\lambda}_a$ denotes the a -th column of $\mathbf{\Lambda}$, $\boldsymbol{\pi}_a$ denotes the a -th column of $\mathbf{\Pi}$, and for two vectors \mathbf{u} and \mathbf{v} with the same dimensions, $\mathbf{u} \odot \mathbf{v}$ denotes the vector obtained through componentwise multiplication. Thus, the decision made at time i by the denoiser of Eq. (2) upon observing a (noisy) sequence \mathbf{z}^n is viewed as a function of the noisy symbol z_i to be corrected. This function depends on the occurrences of a *left context* z_{i-k}^{i-1} , denoted $\mathbf{s}_\ell^{(i)}$, and a *right context* z_{i+1}^{i+k} , denoted $\mathbf{s}_r^{(i)}$, in \mathbf{z}^n (as well as on the parameters $\mathbf{\Pi}$ and $\mathbf{\Lambda}$ of the system).² For a given pair of contexts $(\mathbf{s}_\ell, \mathbf{s}_r)$, this function will be denoted by $g_{\mathbf{z}^n, (\mathbf{s}_\ell, \mathbf{s}_r)}^*$.

It is shown in [9] that for every underlying sequence \mathbf{x}^n , the above denoiser is guaranteed to attain asymptotically, with high probability, the performance of the *best* k -th order sliding-window denoiser, tuned to \mathbf{x}^n and to the observed noisy sequence \mathbf{z}^n . Moreover, for an appropriate growth of k with n , this result holds in an almost sure sense, and in a stochastic setting where the clean sequence is drawn from an unknown stationary source and the competition is with the

²The denoiser output for $i \leq k$ and $i > n-k$, which is (asymptotically) inconsequential, can be assumed to be given by, e.g., an arbitrary symbol.

optimal distribution-dependent denoiser. These results provide asymptotic guidance on the choice of the context length k for universal denoising. However, they refer to a sequence of problems, shedding little light on how k ought to be selected upon observation of a specific sequence \mathbf{z}^n . While we would like to select the “best” value of k given \mathbf{z}^n , our goal cannot be to determine the denoiser in the family that minimizes the loss, as this loss depends on the unobserved sequence \mathbf{x}^n . Instead, we propose in this paper to minimize an *estimate* of the actual loss, that depends on \mathbf{z}^n only. This minimization will be performed over a family of denoisers larger than the one specified in Eq. (2). In the extended family, the context length depends not only on \mathbf{z}^n , but may vary from location to location. Moreover, the context length need not be equal on the left and on the right. Thus, we will minimize the loss estimate over all bi-directional context sets of the type introduced in Section II (up to a preset maximal context length) for the decision rule of Eq. (2), where z_{i-k}^{i-1} and z_{i+1}^{i+k} are replaced by a left context $\mathbf{s}_\ell^{(i)}$ and a right context $\mathbf{s}_r^{(i)}$, respectively, which are determined by the context set and by corresponding left and right directional subsequences $\mathbf{y}_\ell^{(i)} = \{z_{i-1}, z_{i-2}, \dots\}$ and $\mathbf{y}_r^{(i)} = \{z_{i+1}, z_{i+2}, \dots\}$. The vector of counts $\mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell^{(i)}, \mathbf{s}_r^{(i)})$ is generalized accordingly, with its β -th component specifying the number of appearances of $\beta \in \mathcal{X}$ in left context $\mathbf{s}_\ell^{(i)}$ and right context $\mathbf{s}_r^{(i)}$ along \mathbf{z}^n .

Our loss estimate is motivated by Theorem IV.1 below. Let

$$L(x_j^m, \mathbf{z}^n) = \sum_{i=j}^m \Lambda(x_i, \hat{x}_i(z_i))$$

denote the cumulative loss incurred between (and including) locations j and m by a denoiser $\{\hat{x}_i(\cdot)\}$ (where the denoising functions $\hat{x}_i(\cdot)$ may depend on \mathbf{z}^n , as in the denoiser of Eq. (2)), upon observing \mathbf{z}^n , when the underlying clean sequence is \mathbf{x}_j^m . Consider the cumulative loss estimate

$$\hat{L}(\mathbf{z}^n, j, m) = \sum_{i=j}^m \sum_{x \in \mathcal{X}} \Pi^{-T}(x, z_i) \sum_{z \in \mathcal{X}} \Lambda(x, \hat{x}_i(z)) \Pi(x, z) \quad (3)$$

where $\hat{x}_i(z)$ denotes the output of the denoiser at time i when the symbol z_i was replaced by the symbol z in \mathbf{z}^n (therefore, for a denoising function that depends on \mathbf{z}^n , the value of z affects the function itself, and not just its argument). Notice that the estimate of Eq. (3) depends on the observed sequence \mathbf{z}^n , but not on the unobserved sequence \mathbf{x}^n .

Theorem IV.1 *For all $j > 0$, $m \geq j$, $n \geq m$, and $\mathbf{x}^n \in \mathcal{X}^n$, every denoiser satisfies*

$$EL(x_j^m, \mathbf{Z}^n) = E\hat{L}(\mathbf{Z}^n, j, m).$$

A result analogous to Theorem IV.1 was presented in [10, Lemma 4.1] for the causal case, in which the denoiser (filter) must make its decision at location i without access to z_{i+1}^n . The theorem states that the *observable* $\hat{L}(\mathbf{z}^n, j, m)$ is an *unbiased estimate* of $EL(x_j^m, \mathbf{z}^n)$. This property motivates the use of $\hat{L}(\mathbf{z}^n, k+1, n-k)$ as the cumulative loss to be minimized over all possible bi-directional context sets with context length upper-bounded by k , for the denoising rule of Eq. (2). For this case we have, for a given context set \mathcal{S} ,

$$\begin{aligned} \hat{L}(\mathbf{z}^n, k+1, n-k) &= \sum_{i=k+1}^{n-k} \sum_{x \in \mathcal{X}} \Pi^{-T}(x, z_i) \sum_{z \in \mathcal{X}} \Pi(x, z) \\ &\quad \Lambda(x, g_{z_{i-1}^n, (\mathbf{s}_\ell^{(i)}, \mathbf{s}_r^{(i)})}^*(z)) \quad (4) \end{aligned}$$

where the context pair $(\mathbf{s}_\ell^{(i)}, \mathbf{s}_r^{(i)})$ is the one determined by \mathbf{z}^n at location i (through the corresponding left and right directional context sequences) for the given context set \mathcal{S} .

Now, to perform this minimization using the context tree pruning algorithm presented in Section III, we need to decompose the loss estimate given in Eq. (4) into a sum of contributions from each context pair $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}$ (namely, the weights $w(\mathbf{s}_\ell, \mathbf{s}_r)$). Letting $\mathcal{I}_{(\mathbf{s}_\ell, \mathbf{s}_r)} = \{i : k+1 \leq i \leq n-k, (\mathbf{s}_\ell^{(i)}, \mathbf{s}_r^{(i)}) = (\mathbf{s}_\ell, \mathbf{s}_r)\}$, (4) takes the form

$$\hat{L}(\mathbf{z}^n, k+1, n-k) = \sum_{(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}} w(\mathbf{s}_\ell, \mathbf{s}_r)$$

where

$$w(\mathbf{s}_\ell, \mathbf{s}_r) = \sum_{i \in \mathcal{I}_{(\mathbf{s}_\ell, \mathbf{s}_r)}} \sum_{x \in \mathcal{X}} \Pi^{-T}(x, z_i) \sum_{z \in \mathcal{X}} \Pi(x, z) \Lambda(x, g_{z_1^{i-1} z z_{i+1}^n, (\mathbf{s}_\ell, \mathbf{s}_r)}^*(z)). \quad (5)$$

While the weights specified in (5) allow the use of the dynamic programming scheme of Section III to determine a minimizing set \mathcal{S}_{opt} , notice that they may depend on the value of symbols z_i such that $i \notin \mathcal{I}_{(\mathbf{s}_\ell, \mathbf{s}_r)}$. This dependency is due to the fact that the function $g_{z_1^{i-1} z z_{i+1}^n, (\mathbf{s}_\ell, \mathbf{s}_r)}^*(\cdot)$ depends on the vector $\mathbf{m}(z_1^{i-1} z z_{i+1}^n, \mathbf{s}_\ell, \mathbf{s}_r)$, whose components may be affected by appearances of the context pair $(\mathbf{s}_\ell, \mathbf{s}_r)$ in the sequence $z_1^{i-1} z z_{i+1}^n$ at locations other than those specified by $\mathcal{I}_{(\mathbf{s}_\ell, \mathbf{s}_r)}$. As discussed in Section III, such a dependency affects the efficiency of the pruning algorithm in its weight-gathering stage.

To overcome this problem, we will use an *approximate* set of weights. Notice that $\mathbf{m}(z_1^{i-1} z z_{i+1}^n, \mathbf{s}_\ell, \mathbf{s}_r)$ differs from $\mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell, \mathbf{s}_r)$ in two possible ways when $z \neq z_i$. First, replacing z_i with z increases the z -th component by 1 and decreases the z_i -th component by 1. Second, such a replacement may induce new appearances (and cancel actual appearances) of the context pair $(\mathbf{s}_\ell, \mathbf{s}_r)$ in the vicinity of location i . The first situation occurs whenever $z \neq z_i$, but it is not problematic as it depends only on the subsequence $\mathbf{z}(\mathbf{s}_\ell, \mathbf{s}_r)$. The second situation is the problematic one, but in practice it will rarely occur, as it requires that the context pair in question overlap with itself over significant portions. Thus, it will usually yield a second order contribution to the weight, and we will disregard it. This approximation yields a new set of weights, given by

$$\tilde{w}(\mathbf{s}_\ell, \mathbf{s}_r) = \sum_{\beta \in \mathcal{X}} \mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell, \mathbf{s}_r)[\beta] \sum_{x \in \mathcal{X}} \Pi^{-T}(x, \beta) \sum_{z \in \mathcal{X}} \Pi(x, z) \Lambda(x, g_{z_1^{i-1} z z_{i+1}^n, (\mathbf{s}_\ell, \mathbf{s}_r)}^{z \setminus \beta}(z)) \quad (6)$$

where the denoising function $g_{z_1^{i-1} z z_{i+1}^n, (\mathbf{s}_\ell, \mathbf{s}_r)}^{z \setminus \beta}(\cdot)$ is defined as $g_{z_1^{i-1} z z_{i+1}^n, (\mathbf{s}_\ell, \mathbf{s}_r)}^*(\cdot)$, but with the vector $\mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell, \mathbf{s}_r)$ replaced with $\mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell, \mathbf{s}_r) + \mathbf{1}_{z \setminus \beta}$, $\mathbf{1}_{z \setminus \beta}$ denoting a vector with z -th component equal to 1, β -th component equal to -1 , and whose all other components are 0, in case $\beta \neq z$, or the all-zero vector otherwise. Clearly, $\tilde{w}(\mathbf{s}_\ell, \mathbf{s}_r)$ depends on \mathbf{z}^n only through $\mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell, \mathbf{s}_r)$, and can be efficiently employed for bi-directional context tree pruning.

The proposed algorithm has been applied to text denoising, improving over the number of errors after denoising reported in [9] by approximately 20%. In [9], a fixed context length of $k = 2$ was used. In addition, experiments performed over

binary data (generated by a first order Markov source) corrupted by a binary symmetric channel show that the loss estimate \hat{L} of Eq. (3) is indeed very close to the actual loss (with the difference being usually less than 1%) for any denoiser in the family defined by Eq. (2), yielding indeed the best choice of k .

ACKNOWLEDGMENTS

We thank Giovanni Motta, Gadiel Seroussi, and Sergio Verdú for useful discussions.

REFERENCES

- [1] J. Rissanen, "A universal data compression system," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 656–664, Sept. 1983.
- [2] M. J. Weinberger, J. Rissanen, and M. Feder, "A universal finite memory source," *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 643–652, May 1995.
- [3] R. E. Krichevskii and V. K. Trofimov, "The performance of universal encoding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 199–207, Mar. 1981.
- [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] R. Nohre, *Some Topics in Descriptive Complexity*. PhD thesis, Department of Computer Science, The Technical University of Linköping, Sweden, 1994.
- [6] A. Martín, G. Seroussi, and M. J. Weinberger, "Linear time universal coding and time reversal of tree sources via FSM closure," *IEEE Trans. Inform. Theory*, vol. IT-50, pp. 1442–1468, July 2004.
- [7] E. Ordentlich, M. J. Weinberger, and T. Weissman, "Multi-directional context sets with applications to universal denoising and compression," in preparation.
- [8] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "Context weighting for general finite-context sources," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 1514–1520, Sept. 1996.
- [9] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, "Universal discrete denoising: Known channel," *Proceedings of IEEE Symp. on Info. Theory*, 2003, p. 84.
- [10] E. Ordentlich, T. Weissman, M. J. Weinberger, A. Baruch-Somekh, and N. Merhav, "Discrete universal filtering through incremental parsing," *Proceedings of the 2004 Data Compression Conference (DCC'04)*, pp. 352–361, Mar. 2004.