On Multi-Directional Context Sets

Erik Ordentlich, *Fellow, IEEE*, Marcelo J. Weinberger, *Fellow, IEEE*, and Cheng Chang

***Abstract*—The classical framework of context-tree models used in sequential decision problems such as compression and prediction is generalized 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 direction. Tree representations of context sets and pruning algorithms for those trees are extended from the uni-directional setting to two directions. We further show that such tree representations do not extend, in general, to m directions, $m > 2$, and that, as a result, determining the best m-directional context set for $m > 2$ may be substantially more complex than in the case of $m \leq 2$. An application of the proposed pruning algorithm to denoising, where $m = 2$, is presented.**

***Index Terms*—Context trees, denoising, dynamic programming, multi-directional context sets, multi-tracked data, tree pruning algorithms.**

I. Introduction

THE classical context modeling technique [1]–[3] 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 finite context set.[1] Additional 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 a finite set of finite length strings of symbols comprising a context set, where $\mathcal{X}^*$ is the set of all finite length strings (including the empty string) over the alphabet $\mathcal{X}$. For $\mathbf{s} \in \mathcal{S}$ define

$$\mathcal{P}(\mathbf{s}) = \{\mathbf{u} \in \mathcal{X}^* : |\mathbf{u}| \geq |\mathbf{s}|,\ \mathbf{u}^{|\mathbf{s}|} = \mathbf{s}\}$$

where $|\mathbf{s}|$ denotes the length of $\mathbf{s}$ and $\mathbf{u}^l$ is the l symbol prefix of $\mathbf{u}$, $l \leq |\mathbf{u}|$. Further, for any given nonnegative integer j, $j \geq |\mathbf{s}|$, define

$$\mathcal{P}_j(\mathbf{s}) = \{\mathbf{u} \in \mathcal{X}^j : \mathbf{u}^{|\mathbf{s}|} = \mathbf{s}\}.$$

A *valid* context set $\mathcal{S}$ must satisfy the following two properties, where k denotes the (finite) length of the longest string in $\mathcal{S}$:

(1) $\cup_{\mathbf{s}\in S}\mathcal{P}_k(\mathbf{s}) = \mathcal{X}^k$ (exhaustive), and

(2) $\mathcal{P}_k(\mathbf{s}) \cap \mathcal{P}_k(\mathbf{s}') = \emptyset$ for any pair $\mathbf{s} \neq \mathbf{s}'$ (disjoint).[2]

Given a data sequence $\mathbf{x}^n = (x_1, x_2, \ldots, x_n)$, the subsequence of symbols associated with a context $\mathbf{s}$ in a valid context set $\mathcal{S}$ consists of those symbols x_i whose preceding symbols satisfy $\mathbf{s} = (x_{i-1}, x_{i-2}, \ldots, x_{i-|\mathbf{s}|})$. 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 "disjoint" 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 "disjoint" 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 this work, we generalize the above classical context modeling 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 direction. For example, with two directions ("left" and "right"), 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, corresponding to a stereo audio system, $\mathcal{I} = \{1, 2, 3, \ldots\}$ 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 $\mathbf{y}_\ell^{(i)} = x_{L,i-1}, x_{L,i-2}, \ldots$ and $\mathbf{y}_r^{(i)} = x_{R,i-1}, x_{R,i-2}, \ldots$. In a second example, corresponding to a digital image compression or processing application, $\mathcal{I} = \{1, 2, 3, \ldots\} \times \{1, 2, 3, \ldots\}$, where throughout $\times$ denotes cartesian product. Here, $x_{i,j}$ represents the pixel value in row i and column j of the image and $\mathbf{y}_\ell^{(i,j)}$ and $\mathbf{y}_r^{(i,j)}$ may be set as follows. Let $\mathcal{Y}^{(i,j)}$ be the set of pixels that appear either in the same row but to the left of (i, j) or in rows above (i, j). Let $\mathcal{Y}_\ell^{(i,j)}$ and $\mathcal{Y}_r^{(i,j)}$ represent the components of a partition of $\mathcal{Y}^{(i,j)}$ into two subsets. We can let $\mathbf{y}_\ell^{(i,j)}$ and $\mathbf{y}_r^{(i,j)}$ respectively correspond to $\mathcal{Y}_\ell^{(i,j)}$ and $\mathcal{Y}_r^{(i,j)}$ under suitable orderings, say based on proximity to (i, j). For example, $\mathcal{Y}_r^{(i,j)}$ could include all pixels occurring on or above a diagonal line of pixels extending up from and to the left of (i, j) with $\mathcal{Y}_\ell^{(i,j)}$ being the complementary subset of $\mathcal{Y}^{(i,j)}$. This would introduce considerably greater modeling flexibility over the prevailing approach, which derives uni-directional contexts based on a certain ordering of the full set $\mathcal{Y}^{(i,j)}$ (see, e.g., [5]). One could more generally partition and order $\mathcal{Y}^{(i,j)}$ into m partitions, thus naturally giving rise to m-directional context sets. The denoising setting of [6] involves $\mathcal{I} = \{1, 2, 3, \ldots\}$ with $\mathbf{y}_\ell^{(i)} = x_{i-1}, x_{i-2}, \ldots$ (past symbols) and $\mathbf{y}_r^{(i)} = x_{i+1}, x_{i+2}, \ldots$ (future symbols).

Manuscript received November 14, 2009; February 02, 2011; accepted June 06, 2011. Date of current version October 07, 2011. The material in this paper was presented in part at the 2004 IEEE Information Theory Workshop, San Antonio, TX, October 2004, and at the 2005 IEEE International Symposium on Information Theory, Adelaide, Australia, September 2005.

E. Ordentlich and M. J. Weinberger are with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: eord@hpl.hp.com; marcelo@hpl.hp.com).

C. Chang was with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA. He is now with D. E. Shaw & Co., L.P., New York, NY 10036 USA (e-mail: chechang@ocf.berkeley.edu).

Communicated by E.-H. Yang, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2011.2165818

[1] In this paper we will always assume finite context sets; the reader is referred to [4] for a discussion on infinite context sets.

[2] Clearly, the exhaustive property can be equivalently formulated with any integer $k' \geq k$, whereas the disjoint property is equivalent to requiring $\mathcal{P}(\mathbf{s}) \cap \mathcal{P}(\mathbf{s}') = \emptyset$ for any pair $\mathbf{s} \neq \mathbf{s}'$.

With such scenarios in mind, this paper starts by formalizing the notion of a valid *multi-directional* context set.

We further show in this paper that the above tree-based representation of a valid uni-directional context set can be generalized to the case of two directions. Such representations are of paramount importance in applications of context models, where the processing of a context-induced subsequence of data symbols results in a numerical, measurable 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 [7]. In a prediction application, the loss might be the prediction error incurred by a sequential prediction algorithm, such as that of Hannan [8], applied to the subsequence. In denoising, the loss might be an *estimate* of the error (with respect to an unobserved "clean" sequence) incurred by a context-based denoiser, such as the DUDE algorithm [6], applied again to the subsequence.[3] 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 that, for a given individual data sequence $\mathbf{x}^n$ (which generates context formation sequences), 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 $\mathcal{X}$.[4] Given λ and an individual data sequence $\mathbf{x}^n$, of interest is that valid context set $\mathcal{S}$ that minimizes $\sum_{\mathbf{s}\in\mathcal{S}} \lambda(\mathbf{x}(\mathbf{s}))$. In the uni-directional case, an efficient dynamic-programming-based algorithm that relies on the tree representation of valid context sets is known for carrying out this computation [9]. The algorithm "prunes" the (large) context tree formed by all contexts occurring in the given sequence, keeping as leaves those nodes that correspond to the optimal context set. 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 [10] 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 $\mathcal{S}_{\rm opt}$ determined via the aforementioned computation, and second of the sequence compressed via arithmetic coding based on symbol probabilities generated by context-dependent KT estimators.

Our generalization to two directions of the above tree-based representation (bi-directional context trees) leads to the notion of a context "split," by which a bi-directional context tree can be interpreted as specifying a recursive sequence of context splits. In [11], weighting and pruning algorithms are proposed for classes of context sets that are generated according to a variety of context splits. By applying the algorithms in [11] to the specific context splits derived in this paper, we obtain an analogue of the efficient algorithm in [9] for determining the best set of 2-directional (bi-directional) contexts for a given individual sequence and subsequence weight function.

Our final result states that, in general, the context splits noted above do *not* apply to m-directional context sets for $m > 2$. In fact, we demonstrate a family of context sets that cannot be obtained by any nontrivial recursive sequence of context splits in which a context $\mathtt{s}$ is split into a number of contexts having $\mathtt{s}$ as prefix. Moreover, this family further demonstrates that no finite set of recursively applied prefix-like context splits suffices to generate all 3-directional context sets. This negative result highlights the nontrivial nature of the representation for $m = 2$. It also implies that determining the best context set for $m > 2$ may be substantially more complex than in the case of $m \leq 2$.

It is well known that uni-directional context sets can also be interpreted as prefix-free source codes. In the case of $m \geq 2$, one can similarly employ multi-directional context sets as prefix-like codes whose constituent tracks are transmitted or stored on distinct parallel channels. This source coding counterpart to multi-directional context sets was recently studied in [12]. We remark that although many of the structural results (e.g., the bi-directional context tree representation of bi-directional context sets) are isomorphic in the two settings, the respective optimization problems on the relevant structures are quite distinct.

The rest of this paper is organized as follows. In Section II, we define the concept of a bi-directional context set and show that any such set admits a tree representation. In Section III, we use this representation to derive a pruning algorithm for obtaining the optimal set for an individual sequence. In Section IV, we consider the case $m > 2$, for which we show that, in general, such a representation does not exist. In Section V, we discuss in greater depth the denoising application, which was the original motivation of this work. Section VI concludes our paper.

II. Bi-Directional Context Sets: Definition and Structure

Paralleling the classical, uni-directional 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. For $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{X}^* \times \mathcal{X}^*$, define

$$\mathcal{P}(\mathbf{s}_\ell, \mathbf{s}_r) = \{(\mathbf{u}, \mathbf{v}) \in \mathcal{X}^* \times \mathcal{X}^* : \mathbf{u}^{|\mathbf{s}_\ell|} = \mathbf{s}_\ell, \mathbf{v}^{|\mathbf{s}_r|} = \mathbf{s}_r\}$$

namely, the set of all the pairs of strings whose first and second components respectively have $\mathbf{s}_\ell$ and $\mathbf{s}_r$ as prefixes. For $\mathtt{t} \in \mathcal{X}^* \times \mathcal{X}^*$, if $\mathtt{s} \in \mathcal{P}(\mathtt{t})$ we say that $\mathtt{t}$ is an *ancestor* of $\mathtt{s}$ (and $\mathtt{s}$ a *descendant* of $\mathtt{t}$). In analogy to the uni-directional case

$$\mathcal{P}_j(\mathbf{s}_\ell, \mathbf{s}_r) = \{(\mathbf{u}, \mathbf{v}) \in \mathcal{X}^j \times \mathcal{X}^j : \mathbf{u}^{|\mathbf{s}_\ell|} = \mathbf{s}_\ell, \mathbf{v}^{|\mathbf{s}_r|} = \mathbf{s}_r\}$$

is the subset of $\mathcal{P}(\mathbf{s}_\ell, \mathbf{s}_r)$ comprising all the strings in which both components have length j, $j \geq \max(|\mathbf{s}_\ell|, |\mathbf{s}_r|)$. Let k be the

[3]The use of an error estimate based only on observed data makes the loss measurable. Such estimates are investigated in [13].

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

length of the longest string in any pair in $\mathcal{S}$. 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) $\cup_{(\mathbf{s}_\ell,\mathbf{s}_r)\in S}\mathcal{P}_k(\mathbf{s}_\ell,\mathbf{s}_r) = \mathcal{X}^k \times \mathcal{X}^k$ (exhaustive);

(2) $\mathcal{P}_k(\mathbf{s}_\ell,\mathbf{s}_r) \cap \mathcal{P}_k(\mathbf{s}'_\ell,\mathbf{s}'_r) = \emptyset$ for any pair $(\mathbf{s}_\ell,\mathbf{s}_r) \neq (\mathbf{s}'_\ell,\mathbf{s}'_r)$ (disjoint).[5]

Mapping finite-length binary strings to subintervals in $[0,1)$, as customary in the arithmetic coding literature, leads to a geometric interpretation of the binary case that will prove useful in providing intuition to the results in this paper. Specifically, a binary string $\mathbf{u}^k$ corresponds to the subinterval $[.u_1u_2\ldots u_k, .u_1u_2\ldots u_k + 2^{-k})$, where $.u_1u_2\ldots u_k$ denotes a base-2 fractional representation (in particular, for $k = 0$, $\emptyset$ corresponds to the entire unit interval). Similarly, a pair $(\mathbf{u}^k,\mathbf{v}^{k'})$ corresponds to the (axis parallel) rectangle $[.u_1u_2\ldots u_k, .u_1u_2\ldots u_k + 2^{-k}) \times [.v_1v_2\ldots v_{k'}, .v_1v_2\ldots v_{k'} + 2^{-k'})$ in the unit square. This rectangle strictly contains all the rectangles corresponding to descendants of $(\mathbf{u}^k,\mathbf{v}^{k'})$, and in fact coincides with the union of all the rectangles corresponding to the elements of $\mathcal{P}(\mathbf{u}^k,\mathbf{v}^{k'})$, as well as with the (disjoint) union of all the rectangles corresponding to the elements of $\mathcal{P}_{k''}(\mathbf{u}^k,\mathbf{v}^{k'})$ for all $k'' \geq \max(k,k')$. Clearly, the exhaustive and disjoint properties imply that a valid context set corresponds to a partition of the unit square into such rectangles.

Given a data sequence $\mathbf{x} = \{x_i : i \in \mathcal{I}\}$ and, for each $i \in \mathcal{I}$, context formation sequences $\mathbf{y}_\ell^{(i)}$ and $\mathbf{y}_r^{(i)}$ (as exemplified in Section I), the subsequence of symbols associated with a context pair $(\mathbf{s}_\ell,\mathbf{s}_r) \in \mathcal{S}$ consists of those symbols x_i with indexes satisfying

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

and

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

As in the classical case, the new "exhaustive" and "disjoint" properties guarantee that each x_i belongs to one and only one such subsequence. For each $(\mathbf{s}_\ell,\mathbf{s}_r) \in \mathcal{S}$, let $\mathbf{x}(\mathbf{s}_\ell,\mathbf{s}_r)$ denote the subsequence of data symbols associated in the above manner with $(\mathbf{s}_\ell,\mathbf{s}_r)$.

The new "exhaustive" and "disjoint" properties also 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 theorem and which we shall refer to as a bi-directional context tree, has a somewhat different structure from the uni-directional case, and is not necessarily unique for a given $\mathcal{S}$.

Theorem 2.1: 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 Theorem 2.1, 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.

Proof of Theorem 2.1: Our proof is by induction on $m(\mathcal{S})$, defined as the maximum sum of the lengths of the context pair components, that is

$$m(\mathcal{S}) = \max_{(\mathbf{s}_\ell,\mathbf{s}_r)\in\mathcal{S}} [|\mathbf{s}_\ell| + |\mathbf{s}_r|].$$

Only $\mathcal{S} = (\emptyset,\emptyset)$ satisfies the base case $m(\mathcal{S}) = 0$, and the theorem clearly holds for this case with a bi-directional context tree consisting only of the root-node/leaf $\{(\emptyset,\emptyset)\}$. Next, consider any valid $\mathcal{S}$ with $m(\mathcal{S}) = \tilde{m} \geq 1$ and assume, by induction, that the theorem holds for all valid $\mathcal{S}'$ with $m(\mathcal{S}') < \tilde{m}$. We show how it follows that the theorem also holds for $\mathcal{S}$.

First, note that $\mathcal{S}$ cannot contain $(\emptyset,\emptyset)$ since there must be at least one other pair $(\mathbf{s}_\ell,\mathbf{s}_r)$ in $\mathcal{S}$ and $\mathcal{P}_k(\emptyset,\emptyset) = \mathcal{X}^k \times \mathcal{X}^k$ could not be disjoint with $\mathcal{P}_k(\mathbf{s}_\ell,\mathbf{s}_r)$, where $k = k(\mathcal{S})$ denotes the length of the longest string in any pair in $\mathcal{S}$. Second, $\mathcal{S}$ cannot contain both a pair of the form $(\mathbf{s}_\ell,\emptyset)$ and $(\emptyset,\mathbf{s}_r)$ with $\mathbf{s}_\ell \neq \emptyset$ and $\mathbf{s}_r \neq \emptyset$ since then $\mathcal{P}_k(\mathbf{s}_\ell,\mathbf{s}_r)$ would be contained in both $\mathcal{P}_k(\mathbf{s}_\ell,\emptyset)$ and $\mathcal{P}_k(\emptyset,\mathbf{s}_r)$, again violating the "disjoint" condition in the definition of a valid $\mathcal{S}$. Thus, we are faced with two possibilities: Either all right contexts of each pair in $\mathcal{S}$ are non-empty strings, or all left contexts are non-empty strings. Assume, without loss of generality, that the former is the case. It is therefore possible to classify the pairs in $\mathcal{S}$ into $|\mathcal{X}|$ disjoint subsets of the form

$$\mathcal{S}'_x = \{(\mathbf{s}_\ell,\mathbf{s}_r) \in \mathcal{S} : \mathbf{s}_r = (x\mathbf{s}'), \mathbf{s}' \in \mathcal{X}^*, |\mathbf{s}'| \leq k\}$$

for each $x \in \mathcal{X}$. The "exhaustive" property of $\mathcal{S}$ implies that $\mathcal{S}'_x$ is non-empty for each $x \in \mathcal{X}$. For each $\mathcal{S}'_x$ define

$$\mathcal{S}_x = \{(\mathbf{s}_\ell,\mathbf{s}_r) : (\mathbf{s}_\ell, x\mathbf{s}_r) \in \mathcal{S}'_x\}.$$

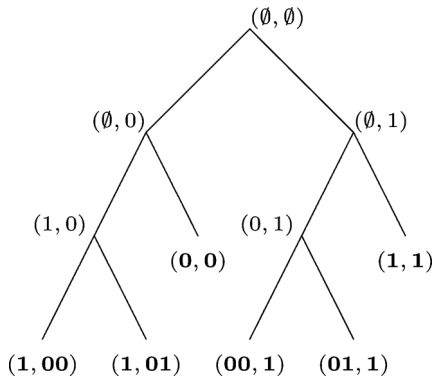
It follows that each $\mathcal{S}_x$ is non-empty (though may consist of the empty pair $(\emptyset,\emptyset)$). Additionally, since $\mathcal{S}$ is valid, it is not hard to see that each $\mathcal{S}_x$ must also be a valid context set. Since $m(\mathcal{S}_x) < m(\mathcal{S}) = \tilde{m}$, the induction hypothesis implies that associated with each $\mathcal{S}_x$ is a bi-directional context tree T_x. For each T_x, let T'_x denote the subtree rooted at $(\emptyset,x)$ obtained by changing each node $(\mathbf{s}_\ell,\mathbf{s}_r)$ of T_x to $(\mathbf{s}_\ell, x\mathbf{s}_r)$ while retaining the parent-child relationships of T_x.

Consider now the tree T consisting of a root node $(\emptyset,\emptyset)$ connected to the subtrees T'_x (i.e., the root-node's children consist of the root-nodes of T'_x, $x \in \mathcal{X}$). The properties of the subtrees T'_x then imply that T is a bi-directional context tree and that its leaves constitute $\cup_{x\in\mathcal{X}}\mathcal{S}'_x = \mathcal{S}$. Since $\tilde{m}$ and $\mathcal{S}$ were arbitrary, the theorem is proved by induction. □

The following is an example of a valid bi-directional context set and an associated bi-directional context tree, as guaranteed by Theorem 2.1. 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.

[5]Again, as in the uni-directional case, there is an equivalent formulation of the disjoint property in terms of $\mathcal{P}(\mathbf{s}_\ell,\mathbf{s}_r)$, which we will occasionally use.

Example 2.2: 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 as shown below.



An equivalent (but different) tree representation can be obtained by letting $(0,\emptyset)$ and $(1,\emptyset)$ be the nodes at level 1 of the tree, $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ (from left to right), be the nodes at level 1 of the tree, and keeping the same sets of children for nodes $(0,1)$ and $(1,0)$.

A bi-directional context tree can be interpreted as specifying, for a valid $\mathcal{S}$ such that the longest string in any pair has length k, a recursive sequence of splits of the set $\mathcal{X}^k \times \mathcal{X}^k$ into the sets $\mathcal{P}_k(\mathbf{s}_\ell, \mathbf{s}_r)$ for $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{S}$. Theorem 2.1 thus shows that valid bi-directional context sets can be obtained as a sequence of splits, where the three possible splits 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 terms of the geometric representation described earlier, Theorem 2.1 states that any partition of the unit square into rectangles corresponding to a valid bi-directional context set can be obtained by a recursive sequence of bisections of rectangles along the axes.

Remark 1: The non-uniqueness of the bi-directional context tree representation introduces a modest redundancy into the description of the optimal context set in the two-part universal source coding application alluded to in the introduction, since trees that represent the same context set are still assigned separate code words. This redundancy can be considered to be of a second-order nature since it amounts to a constant number of bits per context (at most $\log_2 3/2$ bits) and one would expect the optimal number of contexts to be fixed or very slowly growing with the sequence length in most applications. Nevertheless, it would be of some interest to obtain a non-redundant representation of bi-directional context sets, particularly one that is compatible with the algorithm for optimizing context sets described in the next section.

Remark 2: A related tree structure called Bidirectional Context Tree (BCT) was independently proposed in [14], [15][6] in the context of probabilistic bi-directional modeling. While bi-directional context sets are not formally defined in [14], [15], the use of the BCT structure in [16], in conjunction with a mixing algorithm [11], suggests the generation of a bi-directional context set through a recursive sequence of splits different from the one described in Theorem 2.1. Specifically, in the BCT-based approach, nodes $(\mathbf{u},\mathbf{v})$ can split in either direction (or in both directions) provided $|\mathbf{u}| = |\mathbf{v}|$, whereas they can split only in the direction of the *longer* string, if $|\mathbf{u}| \neq |\mathbf{v}|$. In contrast to the statement of Theorem 2.1, it is easy to see that such a sequence of splits cannot generate *every* bi-directional context set. A simple counterexample is given by the set $\{(\emptyset,0),(0,1),(1,1)\}$. However, such restricted sequences of splits may be appropriate for certain specific applications (see Section V).

[6]Notice that [14] is contemporaneous with the conference versions of this paper.

III. Bi-Directional Pruning

Applications similar to those described in Section I for uni-directional context sets motivate the computation of the best bi-directional context set for a given individual sequence. In [11], weighting and pruning algorithms are proposed for classes of context sets that are generated according to a variety of context splits. While the context splits relevant to bi-directional context sets described in Section II are not specifically considered in [11], it is straightforward to extend to this case the algorithms in [11] for finding optimal context sets (and for weighting among these sets in compression applications). For completeness, we describe such an algorithm, based on dynamic programming, next.

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

$$L(\mathcal{S}) = \sum_{(\mathbf{s}_\ell,\mathbf{s}_r)\in\mathcal{S}} \lambda(\mathbf{x}(\mathbf{s}_\ell,\mathbf{s}_r)). \tag{1}$$

Of interest is

$$\mathcal{S}_{\text{opt}} = \arg \min_{\text{valid } \mathcal{S}\subseteq\mathcal{X}^{0:k}\times\mathcal{X}^{0:k}} L(\mathcal{S})$$

where $\mathcal{X}^{0:k}$ is the set of strings over $\mathcal{X}$ of length at most k, and ties are broken according to a deterministic but arbitrary rule. In words, we are seeking a valid context set $\mathcal{S}$, whose contexts have length at most k, that minimizes the loss for $\mathbf{x}^n$.[7]

For any pair $(\mathbf{s}_\ell,\mathbf{s}_r) \in \mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$ we define the weight of $(\mathbf{s}_\ell,\mathbf{s}_r)$ as

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

i.e., 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. By (1), 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 Theorem 2.1, let $w(T)$ denote the weight of T defined as the sum of the weights of the leaves. Let $\mathcal{T}_{\text{opt}}$ be the set of bi-directional context trees with nodes in $\mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$ having minimal weight.

[7]The context length limitation is not critical since, in principle, the algorithm can be applied with $k = n$. In the uni-directional case, this idea can be efficiently implemented by use of compact suffix trees (see [10]). In the bi-directional case, an analogous data structure, the compact bi-directional context graph, is introduced in [17], which combined with algorithms for finding lowest common ancestors yields an efficient pruning algorithm.

In general, this set will have cardinality greater than one, even when $\mathcal{S}_{\rm opt}$ is unique, due to the multiple representations of the context set. Theorem 2.1 and the above observations then imply that $\mathcal{S}_{\rm opt}$ corresponds to the leaves of an element of $\mathcal{T}_{\rm opt}$, thereby reducing the problem of determining $\mathcal{S}_{\rm opt}$ to the problem of determining an element of $\mathcal{T}_{\rm opt}$.

Given any pair $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$, a bi-directional context subtree rooted at $(\mathbf{s}_\ell, \mathbf{s}_r)$ has nodes in $\mathcal{P}(\mathbf{s}_\ell, \mathbf{s}_r)$ 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 $(\mathbf{s}_\ell, \mathbf{s}_r)$ such that every node is in $\mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$. Extend the definition of the weight function $w(T)$ to bi-directional subtrees T in the obvious way and let $\mathcal{T}_{\rm opt}(\mathbf{s}_\ell, \mathbf{s}_r)$ 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, whose proof follows from [11].

Lemma 3.1: For any pair $(\mathbf{s}_\ell, \mathbf{s}_r) \in \mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$

$$\min_{T \in \mathcal{T}(\mathbf{s}_\ell, \mathbf{s}_r)} w(T) = \min \left[w(\mathbf{s}_\ell, \mathbf{s}_r), \quad \sum_{x \in \mathcal{X}} \min_{T' \in \mathcal{T}(\mathbf{s}_\ell x, \mathbf{s}_r)} w(T'), \quad \sum_{x \in \mathcal{X}} \min_{T' \in \mathcal{T}(\mathbf{s}_\ell, \mathbf{s}_r x)} w(T') \right] \quad (2)$$

where we take $\mathcal{T}(\tilde{\mathbf{s}}_\ell, \tilde{\mathbf{s}}_r)$ to be empty if $(\tilde{\mathbf{s}}_\ell, \tilde{\mathbf{s}}_r) \notin \mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$ and the minimum of any function over an empty set to be infinity. If the minimum on the right-hand side of (2) is achieved by the first argument, then the tree $\{(\mathbf{s}_\ell, \mathbf{s}_r)\} \in \mathcal{T}_{\rm opt}(\mathbf{s}_\ell, \mathbf{s}_r)$. Otherwise,

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

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

Lemma 3.1 parallels a similar principle of optimality for the uni-directional case (where the minimum in (2) is over *two* values), and suggests the following dynamic programming algorithm for determining an element of $\mathcal{T}_{\rm opt}$.

Algorithm 3.2

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

We shall refer to Algorithm 3.2 as carrying out a bi-directional context pruning. Although the output of the algorithm is a bi-directional context tree, it should be noticed that the data structure being pruned is *not* a tree [17]. In the algorithm, $T_{\rm opt}(\mathbf{s}_\ell, \mathbf{s}_r)$ is taken to be empty for $(\mathbf{s}_\ell, \mathbf{s}_r) \notin \mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$ and the weight of an empty tree is taken to be infinity. The following theorem, which follows from Theorem 2.1 and Lemma 3.1, establishes that Algorithm 3.2 carries out the desired computation.

Theorem 3.3: The tree $T_{\rm opt}(\emptyset, \emptyset)$ generated by Algorithm 3.2 is an element of $\mathcal{T}_{\rm opt}$ and its leaves constitute $\mathcal{S}_{\rm opt}$.

In many applications, such as two-part codes with KT estimation [10] and denoising using loss estimates (see Section V), $w(\mathbf{s}_\ell, \mathbf{s}_r)$ is a relatively simple function of the vector of counts

$$\mathbf{m}(\mathbf{x}^n, \mathbf{s}_\ell, \mathbf{s}_r)[x] = \sum_{i: x_i \in \mathbf{x}(\mathbf{s}_\ell, \mathbf{s}_r)} 1(x_i = x) \quad (3)$$

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 $|\mathbf{s}_\ell| + |\mathbf{s}_r| < 2k$

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

Finally, we note that the complexity of Algorithm 3.2 can be reduced by restricting processing only to those contexts that actually occur in the sequence (see [17] for a more efficient implementation).

IV. The General Multi-Directional Case

An m-directional context set consists of m-tuples of strings from $\mathcal{X}^*$. The notion of a valid context set, expressed in terms of the "exhaustive" and "disjoint" properties in Section II, assuming the obvious generalization of $\mathcal{P}(\cdot)$ and $\mathcal{P}_k(\cdot)$, extends readily to $m > 2$. The geometric interpretation also extends, so that a valid m-directional context set corresponds to a partition of a unit cube in m-dimensional space into m-dimensional prisms.

It seems natural to conjecture that Theorem 2.1 continues to hold with an m-directional context tree that is the obvious generalization of the $m = 2$ case defined in the theorem. Such a tree would have m-tuples of strings as nodes. Its root would be the null m-tuple and each node would be a leaf, or have as its children the m-tuples obtained by appending, in turn, all symbols in $\mathcal{X}$ to any single component of the m-tuple determining the node. Thus, for $m = 3$ and $\mathcal{X} = \{0, 1\}$, an

internal node might be $(0, 00, 000)$ with possible child sets $\{(00, 00, 000), (01, 00, 000)\}$, $\{(0, 000, 000), (0, 001, 000)\}$, or $\{(0, 00, 0000), (0, 00, 0001)\}$. It is easy to see, however, that this conjecture is not true. Indeed, consider the set of triples

$$\{(0, 0, \emptyset), (1, \emptyset, 0), (\emptyset, 1, 1), (1, 0, 1), (0, 1, 0)\}. \quad (4)$$

This set is a valid 3-directional context set, as the "exhaustive" and "disjoint" conditions can be seen to hold. To see that this set cannot be represented as the leaves of a 3-directional context tree as described in Section II, note that in such a tree the children of the root node are either $\{(\emptyset, \emptyset, 0), (\emptyset, \emptyset, 1)\}$, $\{(\emptyset, 0, \emptyset), (\emptyset, 1, \emptyset)\}$, or $\{(0, \emptyset, \emptyset), (1, \emptyset, \emptyset)\}$. Each case, however, splits at least one of the three subsets of string triples $\mathcal{P}(0, 0, \emptyset)$, $\mathcal{P}(1, \emptyset, 0)$, or $\mathcal{P}(\emptyset, 1, 1)$, respectively.

While the above counterexample demonstrates that the conjectured context splits cannot generate all valid 3-directional context sets, it is conceivable that a different, fixed and *finite* set of context splits based on prefixes can. For example, if the counterexample were an isolated case, one could think of augmenting the set of possible splits with one that splits a context into 5 contexts, following precisely the pattern of (4) (namely, obtain the first context by extending the first two components with a 0, the second by extending the first component with a 1 and the third with a 0, and so forth).

In this section, we show that no such finite set of context splits exists in the 3-directional case. We do it by demonstrating an *arbitrarily large* family of context sets $\mathcal{S}$ such that no recursive sequence of context splits into less than $|\mathcal{S}|$ subsets, in which a context is split into a number of descendant contexts,[8] can generate $\mathcal{S}$. In other words, only the trivial process of splitting the root node into $|\mathcal{S}|$ subsets generates $\mathcal{S}$. Thus, for $m > 2$, any $\mathcal{X}$, and any k, the collection of valid m-directional context sets is richer than what can be represented by the recursive application of any finite set of such context splits. Notice that the complexity of the corresponding pruning algorithm (with the above restriction on splitting based on prefixes only) is determined, to a large extent, by the number of possible context splits applicable at each node, which the constructed family of examples shows to be necessarily growing doubly exponentially in k.

The construction of the above family of 3-directional context sets for $\mathcal{X} = \{0, 1\}$ requires some additional definitions. In the sequel, a context s denotes a triple $(\mathbf{s}_x, \mathbf{s}_y, \mathbf{s}_z)$ of strings in $\mathcal{X}^*$.

Definition 4.1: We say that context $\mathsf{s} \in \mathcal{X}^{0:k} \times \mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$ can be L-split into contexts $\mathsf{s}'_1, \mathsf{s}'_2, \ldots, \mathsf{s}'_L \in \mathcal{X}^{0:k} \times \mathcal{X}^{0:k} \times \mathcal{X}^{0:k}$, $L > 1$, if and only if
(1) $\mathcal{P}_k(\mathsf{s}) = \bigcup_{i=1}^{L} \mathcal{P}_k(\mathsf{s}'_i)$;
(2) $\mathcal{P}(\mathsf{s}'_i) \bigcap \mathcal{P}(\mathsf{s}'_j) = \emptyset$ if $i \neq j$.

In the above counterexample, $(\emptyset, \emptyset, \emptyset)$ is 5-split into $(0, 0, \emptyset), (1, \emptyset, 0), (\emptyset, 1, 1), (1, 0, 1), (0, 1, 0)$.

Definition 4.2: A valid context set $\mathcal{S}$ is L-splittable if the contexts in $\mathcal{S}$ can be represented as the leaves of a rooted tree having nodes in $\mathcal{X}^* \times \mathcal{X}^* \times \mathcal{X}^*$, where the root node is $(\emptyset, \emptyset, \emptyset)$ and each non-leaf node s is ℓ_{s}-split, where $\ell_{\mathsf{s}} \leq L$ for any non-leaf node s.

[8]Here we assume the obvious generalization of the definition of descendant context from the beginning of Section II.

Clearly, a valid context set $\mathcal{S}$ is always $|\mathcal{S}|$-splittable. Thus, the context set in the above counterexample is 5-splittable, but it can be checked that it is not 4-splittable. Also, by Theorem 2.1, any bi-directional context set is $|\mathcal{X}|$-splittable (with an obvious extension of definitions 4.1 and 4.2 to the case $m = 2$).

The main result of this section builds on the following three observations, where $\mathsf{s}, \mathsf{s}', \mathsf{s}'' \in \mathcal{X}^* \times \mathcal{X}^* \times \mathcal{X}^*$ are 3-directional contexts, and we remind the reader that s is an ancestor of s' if $\mathsf{s}' \in \mathcal{P}(\mathsf{s})$.

Fact 1: If $\mathsf{s}'' \in \mathcal{P}(\mathsf{s}')$ and $\mathsf{s}' \in \mathcal{P}(\mathsf{s})$, then $\mathsf{s}'' \in \mathcal{P}(\mathsf{s})$ (transitivity of ancestry).

Fact 2: Let t denote the *longest common ancestor* of s' and s'', defined as the 3-directional context formed by taking, for each component, the longest common prefixes of the corresponding components of s' and s''. If $\mathsf{s}' \in \mathcal{P}(\mathsf{s})$ and $\mathsf{s}'' \in \mathcal{P}(\mathsf{s})$ then $\mathsf{t} \in \mathcal{P}(\mathsf{s})$.

Fact 3: Let the context set $\mathcal{S}$ be L-splittable, let $\mathsf{s} \in \mathcal{S}$, and let s' be a node on the corresponding tree. If $\mathcal{P}(\mathsf{s}) \bigcap \mathcal{P}(\mathsf{s}') \neq \emptyset$ then $\mathsf{s} \in \mathcal{P}(\mathsf{s}')$.

Facts 1 and 2 are straightforward consequences of the ancestry relationship, whereas Fact 3 clearly follows from the disjointness of a valid context set and the tree structure defining an L-splittable set.

Next, we construct a 3-directional context set $\mathcal{S}$ from three valid, arbitrary uni-directional context sets, $\{\mathbf{s}_x^{(1)}, \ldots, \mathbf{s}_x^{(X)}\}$, $\{\mathbf{s}_y^{(1)}, \ldots, \mathbf{s}_y^{(Y)}\}$, and $\{\mathbf{s}_z^{(1)}, \ldots, \mathbf{s}_z^{(Z)}\}$, each of length less than some fixed integer k. To this end, we arbitrarily pick three collections of binary symbols $\{b_x^{(1)}, \ldots, b_x^{(X)}\}$, $\{b_y^{(1)}, \ldots, b_y^{(Y)}\}$, and $\{b_z^{(1)}, \ldots, b_z^{(Z)}\}$, and let

$$\begin{aligned}
\mathcal{S} = \; & \{\mathbf{s}_x^{(1)} b_x^{(1)}, \ldots, \mathbf{s}_x^{(X)} b_x^{(X)}\} \times \{\mathbf{s}_y^{(1)} \bar{b}_y^{(1)}, \ldots, \mathbf{s}_y^{(Y)} \bar{b}_y^{(Y)}\} \times \emptyset \\
\bigcup \; & \emptyset \times \{\mathbf{s}_y^{(1)} b_y^{(1)}, \ldots, \mathbf{s}_y^{(Y)} b_y^{(Y)}\} \times \{\mathbf{s}_z^{(1)} \bar{b}_z^{(1)}, \ldots, \mathbf{s}_z^{(Z)} \bar{b}_z^{(Z)}\} \\
\bigcup \; & \{\mathbf{s}_x^{(1)} \bar{b}_x^{(1)}, \ldots, \mathbf{s}_x^{(X)} \bar{b}_x^{(X)}\} \times \emptyset \times \{\mathbf{s}_z^{(1)} b_z^{(1)}, \ldots, \mathbf{s}_z^{(Z)} b_z^{(Z)}\} \\
\bigcup \; & \{\mathbf{s}_x^{(1)} b_x^{(1)}, \ldots, \mathbf{s}_x^{(X)} b_x^{(X)}\} \times \{\mathbf{s}_y^{(1)} b_y^{(1)}, \ldots, \mathbf{s}_y^{(Y)} b_y^{(Y)}\} \times \\
& \{\mathbf{s}_z^{(1)} b_z^{(1)}, \ldots, \mathbf{s}_z^{(Z)} b_z^{(Z)}\} \\
\bigcup \; & \{\mathbf{s}_x^{(1)} \bar{b}_x^{(1)}, \ldots, \mathbf{s}_x^{(X)} \bar{b}_x^{(X)}\} \times \{\mathbf{s}_y^{(1)} \bar{b}_y^{(1)}, \ldots, \mathbf{s}_y^{(Y)} \bar{b}_y^{(Y)}\} \times \\
& \{\mathbf{s}_z^{(1)} \bar{b}_z^{(1)}, \ldots, \mathbf{s}_z^{(Z)} \bar{b}_z^{(Z)}\} \qquad (5)
\end{aligned}$$

where $\bar{b}$ denotes the complement of $b \in \{0, 1\}$. The length of each component of $\mathcal{S}$ is therefore at most k. Notice that the counterexample (4) is a particular case of $\mathcal{S}$, where the three uni-directional context sets are the empty set (namely, $X = Y = Z = 1$), $b_x^{(1)} = 0$, $b_y^{(1)} = 1$, and $b_z^{(1)} = 0$.

Theorem 4.3: The 3-directional context set $\mathcal{S}$ is valid and is not $(|\mathcal{S}| - 1)$-splittable.

Discussion: In general, $|\mathcal{S}| = 2XYZ + XY + YZ + ZX$ and for each arbitrary choice of the uni-directional context set triple there are 2^{X+Y+Z} such 3-directional context sets. In particular, X,Y,Z could be as large as 2^{k-1}, and hence the number of context sets $\mathcal{S}$ in the constructed family grows doubly exponentially in k. This fact and Theorem 4.3 readily imply that for any fixed, finite set of context splits based on descendants, for all sufficiently large k, there will exist a context set $\mathcal{S}$ that cannot be obtained via successive applications of the given context splits.

Moreover, we also have that the number of such exceptional context sets grows doubly exponentially in k, for any fixed, finite set of context splits. Note that these conclusions extend to descendant-based context splits that can be a function of the set being split, as long as the range of this function (i.e., the number of possible splits) at each point does not grow (or is sufficiently slowly growing) with k.

Proof of Theorem 4.3: First, we show that $\mathcal{S}$ is valid. Due to the structure of the union of cartesian products forming $\mathcal{S}$ in (5), we have

$$\mathcal{S} = \bigcup_{p,q,r} \mathcal{S}(p,q,r) \tag{6}$$

where, for indexes $p \in \{1,2,\ldots,X\}$, $q \in \{1,2,\ldots,Y\}$, and $r \in \{1,2,\ldots,Z\}$

$$\begin{aligned}\mathcal{S}(p,q,r) = \Big\{ &\left(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset\right), \left(\emptyset, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} \overline{b}_z^{(r)}\right), \\ &\left(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \emptyset, \mathbf{s}_z^{(r)} b_z^{(r)}\right), \left(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} b_z^{(r)}\right), \\ &\left(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \mathbf{s}_z^{(r)} \overline{b}_z^{(r)}\right)\Big\}. \end{aligned} \tag{7}$$

To prove disjointness, we observe that (6) implies that, for any pair of contexts $\mathsf{s}_1, \mathsf{s}_2 \in \mathcal{S}$, there exists a component w (where w is either x, y, or z) such that the corresponding components of s_1 and s_2 are $\mathbf{s}_w^{(i)} b_w^{(i)}$ and $\mathbf{s}_w^{(j)} \overline{b}_w^{(j)}$, for some pair of indexes i, j. The property then follows from the fact that such strings cannot be prefixes of each other due to the disjointness of the corresponding uni-directional context set (if $i \neq j$) or to the discrepancy in the last bit (if $i = j$).

As for the exhaustiveness, notice that since the given uni-directional context sets are valid, for all $(\mathbf{s}_x, \mathbf{s}_y, \mathbf{s}_z) \in \mathcal{X}^k \times \mathcal{X}^k \times \mathcal{X}^k$ there exists a unique triple p, q, r such that $\mathbf{s}_x \in \mathcal{P}_k(\mathbf{s}_x^{(p)})$, $\mathbf{s}_y \in \mathcal{P}_k(\mathbf{s}_y^{(q)})$, and $\mathbf{s}_z \in \mathcal{P}_k(\mathbf{s}_z^{(r)})$. It is then easy to verify that $(\mathbf{s}_x, \mathbf{s}_y, \mathbf{s}_z) \in \bigcup_{\mathsf{s} \in \mathcal{S}(p,q,r)} \mathcal{P}_k(\mathsf{s})$.

Next, suppose that $\mathcal{S}$ is $(|\mathcal{S}|-1)$-splittable. Then, there exists an *intermediate* node (i.e., not root *or* leaf) t in the resulting tree, namely $\mathsf{t} \neq (\emptyset, \emptyset, \emptyset)$ and $\mathsf{t} \notin \mathcal{S}$, such that t can be L-split (possibly in multiple steps) into a subset of contexts $\mathsf{s}_1, \ldots, \mathsf{s}_L \in \mathcal{S}$, $1 < L < |\mathcal{S}|$. Hence, $\mathsf{s}_1 \in \mathcal{P}(\mathsf{t})$ and $\mathsf{s}_1 \neq \mathsf{t}$.

By (6) and the symmetries in the right-hand side of (7), we can assume, without loss of generality, that s_1 takes the form of either $(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset)$ or $(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} b_z^{(r)})$, for some integers p, q, r. We show that, in both cases, $\mathcal{P}(\mathsf{t})$ has to contain $(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset)$, $(\emptyset, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} \overline{b}_z^{(r)})$, and $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \emptyset, \mathbf{s}_z^{(r)} b_z^{(r)})$, and thus, by Fact 2, t must necessarily be $(\emptyset, \emptyset, \emptyset)$, leading to a contradiction.

Case 1: $\mathsf{s}_1 = (\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset)$

Since $\mathsf{s}_1 \in \mathcal{P}(\mathsf{t})$ and $\mathsf{s}_1 \neq \mathsf{t}$, then $\mathcal{P}(\mathsf{t})$ must contain at least one of its ancestors, $(\mathbf{s}_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset)$ or $(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)}, \emptyset)$. Assume that $(\mathbf{s}_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset) \in \mathcal{P}(\mathsf{t})$ (the other case is omitted as it can be treated similarly). Since $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \mathbf{s}_z^{(r)} b_z^{(r)})$ is, in turn, in $\mathcal{P}((\mathbf{s}_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset))$, by Fact 1, it is also in $\mathcal{P}(\mathsf{t})$. Moreover, its ancestor $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \emptyset, \mathbf{s}_z^{(r)} b_z^{(r)})$ is, by (6) and (7), in $\mathcal{S}$, so $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \emptyset, \mathbf{s}_z^{(r)} b_z^{(r)}) \in \mathcal{P}(\mathsf{t})$ according to Fact 3. Now, we have $(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset) \in \mathcal{P}(\mathsf{t})$ and $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \emptyset, \mathbf{s}_z^{(r)} b_z^{(r)}) \in \mathcal{P}(\mathsf{t})$ which, by Fact 2, implies that $(\mathbf{s}_x^{(p)}, \emptyset, \emptyset) \in \mathcal{P}(\mathsf{t})$. In turn, this implies that its descendant $(\mathbf{s}_x^{(p)}, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} \overline{b}_z^{(r)})$, which is also a descendant of the context $(\emptyset, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} \overline{b}_z^{(r)})$ in $\mathcal{S}$, is in $\mathcal{P}(\mathsf{t})$. Thus, again by Fact 3, $(\emptyset, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} \overline{b}_z^{(r)}) \in \mathcal{P}(\mathsf{t})$, leading to the desired contradiction.

Case 2: $\mathsf{s}_1 = (\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} b_z^{(r)})$

Since $\mathsf{s}_1 \in \mathcal{P}(\mathsf{t})$ and $\mathsf{s}_1 \neq \mathsf{t}$, then one of the following three contexts must belong to $\mathcal{P}(\mathsf{t})$: $(\mathbf{s}_x^{(p)}, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} b_z^{(r)})$, $(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)}, \mathbf{s}_z^{(r)} b_z^{(r)})$, or $(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)})$. Noticing the symmetry, we assume that $(\mathbf{s}_x^{(p)}, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} b_z^{(r)}) \in \mathcal{P}(\mathsf{t})$. By Fact 1, its descendant $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} b_z^{(r)}) \in \mathcal{P}(\mathsf{t})$. This context has an ancestor $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \emptyset, \mathbf{s}_z^{(r)} b_z^{(r)}) \in \mathcal{S}$. Thus, by Fact 3, $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \emptyset, \mathbf{s}_z^{(r)} b_z^{(r)}) \in \mathcal{P}(\mathsf{t})$. Following the same argument as in Case 1, we have $(\mathbf{s}_x^{(p)} b_x^{(p)}, \mathbf{s}_y^{(q)} \overline{b}_y^{(q)}, \emptyset) \in \mathcal{P}(\mathsf{t})$, $(\mathbf{s}_x^{(p)} \overline{b}_x^{(p)}, \emptyset, \mathbf{s}_z^{(r)} b_z^{(r)}) \in \mathcal{P}(\mathsf{t})$, and $(\emptyset, \mathbf{s}_y^{(q)} b_y^{(q)}, \mathbf{s}_z^{(r)} \overline{b}_z^{(r)}) \in \mathcal{P}(\mathsf{t})$, leading again to the desired contradiction.

Since both cases lead to a contradiction, namely that t has to be the root node, we conclude that the context set $\mathcal{S}$ is not $(|\mathcal{S}|-1)$-splittable. □

Comments on the Proof: Fig. 1 depicts, according to our geometric interpretation, the constituent contexts (7) in the characterization (6) of the 3-directional context set $\mathcal{S}$ of (5). The figure depicts the prisms corresponding to the five contexts in (7) for a generic choice of the uni-directional contexts $\mathbf{s}_x^{(p)}$, $\mathbf{s}_y^{(q)}$, and $\mathbf{s}_z^{(r)}$ (which correspond, respectively, to the x, y, and z axis projections of the sides of the $(2 \text{ quadrant} \times 2 \text{ quadrant})$ in-plane squares bounding the long prisms, as shown in the figure) and the choice $b_x^{(p)} = 0, b_y^{(q)} = 1$, and $b_z^{(r)} = 1$. Note that the small prisms $(\mathbf{s}_x^{(p)} 1, \mathbf{s}_y^{(q)} 0, \mathbf{s}_z^{(r)} 0)$ and $(\mathbf{s}_x^{(p)} 0, \mathbf{s}_y^{(q)} 1, \mathbf{s}_z^{(r)} 1)$ look like cubes in the figure and would be if the strings $\mathbf{s}_x^{(p)}$, $\mathbf{s}_y^{(q)}$, and $\mathbf{s}_z^{(r)}$ were of equal length, but they need not be cubes in general. As expressed by (6), the context set $\mathcal{S}$ corresponds to a decomposition of the cube $[0,1) \times [0,1) \times [0,1)$ into prisms that can be grouped into structures of the sort depicted in the figure, where each such structure corresponds to a different $\mathcal{S}(p,q,r)$ in (6). Note that a given (long) prism in the decomposition is actually a part of many such structures, since the sets $\mathcal{S}(p,q,r)$ are not disjoint.

We can interpret the proof of Theorem 4.3 in terms of Fig. 1 as follows. Under our geometric interpretation, if a nontrivial tree-like split exists, the prism corresponding to any ancestor $\mathsf{t} \in \{0,1\}^{[0:k]} \times \{0,1\}^{[0:k]} \times \{0,1\}^{[0:k]}$ of one of the five contexts in (7), must strictly contain the prism corresponding to that context, denoted s_1. With respect to the case depicted in the figure (with the specific choices of $b_x^{(p)}, b_y^{(q)}, b_z^{(r)}$), the proof shows that for each of the five prisms, the only strictly containing prism corresponding to an ancestor t (of the corresponding context s_1) that does not also cut any of the three long prisms,[9] is the entire cube, which corresponds to $(\emptyset, \emptyset, \emptyset)$. Note that there do exist prisms that strictly contain one of the prisms in the figure and do not cut any of the long prisms. Consider, for example, the one containing the prism corresponding to $(\mathbf{s}_x^{(p)} 1, \emptyset, \mathbf{s}_z^{(r)} 1)$ and extending to the unit-cube boundaries in the increasing x and z directions. This prism, however, does not correspond to an ancestor of $(\mathbf{s}_x^{(p)} 1, \emptyset, \mathbf{s}_z^{(r)} 1)$.

[9]Notice that cutting a prism corresponds to splitting the corresponding context in $\mathcal{S}$ and therefore does not lead to a valid ancestor.

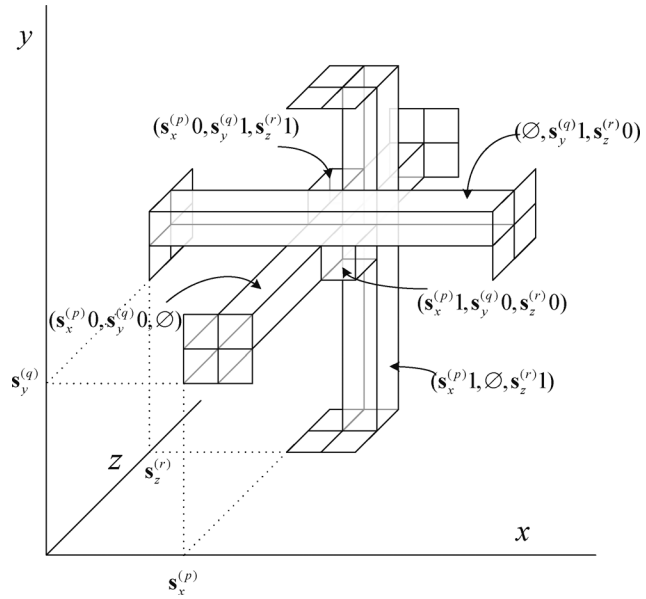


Fig. 1. A geometric interpretation of the constituent contexts (7) in the characterization (6) of the 3-directional context set $\mathcal{S}$ of Theorem 4.3 (defined by (5)).

The intuition behind the existence of a tree decomposition for the bi-directional case is that interwoven structures of prisms of the sort depicted in Fig. 1 are not possible in two dimensions.

V. Application to Denoising

The bi-directional context set framework described in Sections II and III can be applied to context-based denoising, in particular to enhance the DUDE algorithm proposed in [6]. This application was the original motivation of this work, and as hinted in Section I, it is less straightforward than compression or prediction. Therefore, we further discuss it in this section.

In the (semi-stochastic) universal denoising setting, we consider an individual sequence $\mathbf{x}^n$, which is corrupted by a discrete 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 any 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 $\Lambda : \mathcal{X}^2 \to [0,\infty)$, 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 $\hat{x}_i$, $1 \le i \le n$. 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 context-based denoisers, parameterized by a nonnegative integer parameter (context length) k, is proposed in [6]. For a given k, the denoiser output $\hat{x}_i^*(z_i)$ at time i for a noisy input z_i, $k+1 \le i \le n-k$, is given by

$$\hat{x}_i^*(z_i) = \arg\min_{\hat{x}\in\mathcal{X}} \mathbf{m}(\mathbf{z}^n, z_{i-k}^{i-1}, z_{i+1}^{i+k})\mathbf{\Pi}^{-1}\left[\boldsymbol{\lambda}_{\hat{x}} \odot \boldsymbol{\pi}_{z_i}\right] \quad (8)$$

where, similar to (3), $\mathbf{m}(\mathbf{z}^n, z_{i-k}^{i-1}, z_{i+1}^{i+k})$ 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 (8) 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).[10] 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 [6] that for every underlying sequence $\mathbf{x}^n$, the above denoiser is guaranteed to attain asymptotically, with probability one, the performance of the *best* k_n-th order sliding-window denoiser, tuned to $\mathbf{x}^n$ and to the observed noisy sequence $\mathbf{z}^n$, provided k_n grows sufficiently slowly with n. These results provide asymptotic guidance on the choice of k in (8) as a function of n, but do not reveal how k ought to be selected upon observation of a specific sequence $\mathbf{z}^n$. The goal of selecting the denoiser in the family that minimizes the loss, which corresponds precisely to the setting of this paper, is unreachable as it depends on the unobserved sequence $\mathbf{x}^n$. It is then natural to minimize, instead, an *estimate* of the actual loss, that depends on $\mathbf{z}^n$ only. Properties of this strategy, as well as suitable loss estimators, are discussed in [13] and [18]. The minimization will in fact be performed over a larger family of denoisers, which is a natural generalization of the one specified in (8). We consider all bi-directional context sets of the type introduced in Section II (up to a preset maximal context length), replacing z_{i-k}^{i-1} and z_{i+1}^{i+k} in the decision rule of (8) with 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 $\mathbf{y}_\ell^{(i)} = \{z_{i-1}, z_{i-2}, \ldots\}$ and $\mathbf{y}_r^{(i)} = \{z_{i+1}, z_{i+2}, \ldots\}$. The vector of counts $\mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell^{(i)}, \mathbf{s}_r^{(i)})$ is as defined in (3). The goal is thus to find the bi-directional context set that minimizes a given loss estimate, to be presented next, for $\mathbf{z}^n$.

Our estimate of 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 (8)), upon observing $\mathbf{z}^n$, is motivated by the prediction-filtering correspondence stated in [19, Theorem 4].[11] Specifically, let

$$\hat{L}(\mathbf{z}^n, j, m) = \sum_{i=j}^{m}\sum_{x\in\mathcal{X}} \Pi^{-T}(x, z_i)\sum_{z\in\mathcal{X}}\Lambda(x, \hat{x}_i(z))\Pi(x,z) \quad (9)$$

where $\hat{x}_i(z)$ denotes the output of the denoiser at time i when the symbol z_i is 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 innermost summation is the expected loss of the denoiser at time i when $x_i = x$, whereas it is easy to see that the expectation of $\Pi^{-T}(x, Z_i)$ is $\mathbf{1}(x = x_i)$. Further intuition on the rationale behind the estimate of (9) is given in [20]. Notice that this estimate

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

[11] In the filtering problem, the denoiser (filter) must make its decision at location i without access to z_{i+1}^n.

depends on the observed sequence $\mathbf{z}^n$, but not on the unobserved sequence $\mathbf{x}^n$.

Now, to minimize the cumulative loss estimate given in (9) over all possible bi-directional context sets with context length upper-bounded by k, for the denoiser of (8), using the context pruning algorithm presented in Section III, we need to decompose the estimate 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

$$\mathcal{I}_{(\mathbf{s}_\ell,\mathbf{s}_r)} = \{i : k+1 \le i \le n-k,\ (\mathbf{s}_\ell^{(i)}, \mathbf{s}_r^{(i)}) = (\mathbf{s}_\ell, \mathbf{s}_r)\}$$

(9) becomes

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

where

$$w(\mathbf{s}_\ell, \mathbf{s}_r) = \sum_{i\in\mathcal{I}_{(\mathbf{s}_\ell,\mathbf{s}_r)}} \sum_{x\in\mathcal{X}} \Pi^{-T}(x, z_i) \sum_{z\in\mathcal{X}} \Pi(x, z) \Lambda(x, g^*_{z_1^{i-1} z z_{i+1}^n, (\mathbf{s}_\ell,\mathbf{s}_r)}(z)). \quad (10)$$

While the weights specified in (10) allow the use of the dynamic programming scheme of Section III to determine a minimizing set $\mathcal{S}_{\rm opt}$, 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 $\mathcal{I}_{(\mathbf{s}_\ell,\mathbf{s}_r)}$. Such a dependency precludes the use of the efficient procedure for weight computation in the pruning algorithm discussed at the end of Section III.

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

$$\tilde{w}(\mathbf{s}_\ell, \mathbf{s}_r) = \sum_{\beta\in\mathcal{X}} \mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell, \mathbf{s}_r)[\beta] \sum_{x\in\mathcal{X}} \Pi^{-T}(x, \beta) \sum_{z\in\mathcal{X}} \Pi(x, z) \Lambda(x, g^{z\backslash\beta}_{\mathbf{z}^n, (\mathbf{s}_\ell,\mathbf{s}_r)}(z)) \quad (11)$$

where the denoising function $g^{z\backslash\beta}_{\mathbf{z}^n, (\mathbf{s}_\ell,\mathbf{s}_r)}(\cdot)$ is defined as $g^*_{\mathbf{z}^n, (\mathbf{s}_\ell,\mathbf{s}_r)}$, but with the vector $\mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell, \mathbf{s}_r)$ replaced with $\mathbf{m}(\mathbf{z}^n, \mathbf{s}_\ell, \mathbf{s}_r) + \mathbf{1}_{z\backslash\beta}$, $\mathbf{1}_{z\backslash\beta}$ 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 pruning.

The proposed algorithm has been applied to the text denoising example in Section VIII-B of [6]. In this example, an English translation of the novel *Don Quixote* obtained from the Project Gutenberg website (http://promo.net/pg/) was artificially corrupted by a hypothetical "typewriter channel" in which each character was flipped independently with probability .05, equiprobably to one of its nearest neighbors in the QWERTY keyboard. This setting resulted in 89087 errors out of a total of 2.3×10^6 characters. The application of the DUDE algorithm of [6] with fixed-length bi-directional contexts of lengths $k = 0$, 1, 2, and 3 resulted in 87762, 53191, 50250, and 73660 denoising errors, respectively. On the other hand, the above proposed algorithm, with variable length bi-directional contexts with maximum context length of $k = 3$, resulted in 39162 denoising errors, a roughly 20% improvement over the best choice of k for the baseline DUDE algorithm of [6]. The overall loss obtained in this experiment is close to the loss of the best denoiser of type (8) among all bi-directional context sets, validating the loss estimation approach. However, this gradually ceases to be the case as k grows (and along with it, the effective search space of context sets). A version of this problem is discussed in [18]. A different approach for denoising with variable length bi-directional contexts, based on a stochastic modeling of the noisy data, is proposed in [15]. This approach does not use the rule (8).

VI. Conclusion

We have formalized the notion of a multi-directional context set as a natural extension to its uni-directional counterpart, with applications in decision problems such as compression, prediction, and denoising. We have shown that there exists a fundamental difference between the bi-directional case and the m-directional case, $m > 2$, in that bi-directional context sets can be generated by a recursive sequence of splits and represented by $|\mathcal{X}|$-ary trees. When paired with the algorithms in [11], this representation leads to an efficient "pruning" algorithm for obtaining the optimal bi-directional context set for a given loss function, thus extending the well-known dynamic programming procedure of [9]. While natural, the tree representation is not straightforward, as highlighted by the fact that, in general, it does not exist for $m > 2$. Moreover, we have demonstrated an arbitrarily large family of 3-directional context sets which cannot be represented with *any* ℓ-ary tree for ℓ smaller than the size of the context set, at least when context splits follow a prefix-like constraint. The constructed family demonstrates that no finite set of recursively applied prefix-like context splits suffices to generate all 3-directional context sets, and that, in fact, the number of exceptional context sets for any such set of context splits grows doubly exponentially in k, the bound on the context length along any direction. This result raises the question of the existence of efficient algorithms for context set optimization (not based on [11]) for $m > 2$. Such algorithms could be used, e.g., to increase the modeling flexibility in the image processing application described in Section I, by providing an

efficient construction of a conditional pixel model according to more than two directions. Finally, we have demonstrated how to apply the proposed pruning algorithm for bi-directional context sets to denoising.

ACKNOWLEDGMENT

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

REFERENCES

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

[2] M. J. Weinberger, J. Rissanen, and M. Feder, "A universal finite memory source," *IEEE Trans. Inf. Theory*, vol. IT-41, no. 5, pp. 643–652, May 1995.

[3] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inf. Theory*, vol. IT-41, no. 5, pp. 653–664, May 1995.

[4] I. Csiszár and Z. Talata, "Context tree estimation for not necessarily finite memory processes, via BIC and MDL," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1007–1016, Mar. 2006.

[5] M. J. Weinberger, J. J. Rissanen, and R. B. Arps, "Applications of universal context modeling to lossless compression of gray-scale images," *IEEE Trans. Image Process.*, vol. 5, pp. 575–586, Apr. 1996.

[6] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. J. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. Inf. Theory*, vol. 51, pp. 5–28, Jan. 2005.

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

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

[9] R. Nohre, "Some Topics in Descriptive Complexity," Ph.D. dissertation, Department of Computer Science, The Technical University of Linkoping, Linkoping, Sweden, 1994.

[10] A. Martín, G. Seroussi, and M. J. Weinberger, "Linear time universal coding and time reversal of tree sources via FSM closure," *IEEE Trans. Inf. Theory*, vol. IT-50, no. 7, pp. 1442–1468, Jul. 2004.

[11] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "Context weighting for general finite-context sources," *IEEE Trans. Inf. Theory*, vol. IT-42, no. 6, pp. 1514–1520, Sep. 1996.

[12] H. Yao and R. W. Yeung, "Zero-error multichannel source coding," in *Proc. 2010 Information Theory Workshop*, Cairo, Egypt, Jan. 2010.

[13] E. Ordentlich, K. Viswanathan, and M. J. Weinberger, "On concentration for denoiser-loss estimators," in *Proc. 2009 IEEE Int. Symp. Information Theory (ISIT'09)*, Seoul, Korea, Jun. 2009.

[14] J. Yu and S. Verdú, "Schemes for bidirectional modeling of discrete stationary sources," in *Proc. 2005 Conf. Information Sciences and Systems*, The Johns Hopkins University, Baltimore, MD, Mar. 16, 2005.

[15] J. Yu and S. Verdú, "Schemes for bidirectional modeling of discrete stationary sources," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4789–4807, Nov. 2006.

[16] J. Yu and S. Verdú, "Universal estimation of erasure entropy," *IEEE Trans. Inf. Theory*, vol. 55, no. 1, pp. 350–357, Jan. 2009.

[17] F. Fernández, A. Viola, and M. J. Weinberger, "Efficient algorithms for constructing optimal bi-directional context sets," in *Proc. 2010 IEEE Data Compression Conf. (DCC'10)*, Snowbird, UT, Mar. 2010, pp. 179–188.

[18] E. Ordentlich, K. Viswanathan, and M. J. Weinberger, "Toward properties of twice-universality in denoising," in *Proc. 2010 IEEE Int. Symp. Information Theory (ISIT'10)*, Austin, TX, Jun. 2010.

[19] T. Weissman, E. Ordentlich, M. J. Weinberger, A. Baruch-Somekh, and N. Merhav, "Universal filtering via prediction," *IEEE Trans. Inf. Theory*, vol. IT-53, no. 4, pp. 1253–1264, Apr. 2007.

[20] T. Moon and T. Weissman, "Discrete denoising with shifts," *IEEE Trans. Inf. Theory*, vol. IT-55, no. 11, pp. 5284–5301, Nov. 2009.

Erik Ordentlich (S'92–M'96–SM'06–F'11) received the S.B. and S.M. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1990, and the Ph.D. degree, also in electrical engineering, from Stanford University, Stanford, CA, in 1996.

He is a Senior Research Scientist in the Information Theory Research Group at Hewlett-Packard Laboratories, Palo Alto, CA. He has been with Hewlett-Packard Laboratories since 1996, with the exception of a period in 1999–2002, when he was with iCompression, Inc., Santa Clara, CA. His work has addressed multiple topics in signal processing and information theory. He is a co-inventor on numerous U.S. Patents and contributed technology to the ISO JPEG 2000 image compression standard.

Dr. Ordentlich was a co-recipient of the 2006 IEEE Communications/Information Theory Societies Joint Paper Award and is a member of Phi Beta Kappa and Tau Beta Pi. He served as Associate Editor for Source Coding for the IEEE TRANSACTIONS ON INFORMATION THEORY from 2007 to 2010.

Marcelo J. Weinberger (M'90–SM'98–F'07) was born in Buenos Aires, Argentina. He received the Electrical Engineer degree from the Universidad de la República, Montevideo, Uruguay, in 1983, and the M.Sc. and D.Sc. degrees from Technion—Israel Institute of Technology, Haifa, Israel, in 1987 and 1991, respectively, both in electrical engineering.

From 1985 to 1992 he was with the Department of Electrical Engineering at Technion, joining the faculty for the 1991–1992 academic year. During 1992–1993 he was a Visiting Scientist at IBM Almaden Research Center, San Jose, California. Since 1993 he has been with Hewlett-Packard Laboratories, Palo Alto, California, where he is a Distinguished Scientist and leads the Information Theory Research group. His research interests include source coding, sequential decision problems, statistical modeling, and image compression. He is a coauthor of the algorithm at the core of the JPEG-LS lossless image compression standard, and was an editor of the standard specification. He also contributed to the coding algorithm of the JPEG2000 image compression standard.

Dr. Weinberger served as an Associate Editor for Source Coding of the IEEE TRANSACTIONS ON INFORMATION THEORY from 1999 to 2002. He is a co-recipient of the 2006 IEEE Communications/Information Theory Societies Joint Paper Award.

Cheng Chang received the B.E. degree from Tsinghua University, Beijing, in 2000, and the Ph.D. degree from University of California at Berkeley in 2007. He is currently a quantitative analyst with the D. E. Shaw Group in New York. In 2008, he spent a year in the Information Theory Research group at HP Labs as a post-doctoral researcher. His research interests include signal processing, control theory, information theory and machine learning.