Arithmetic Encoding of Markov Random Fields

Matthew G. Reyes and David L. Neuhoff
EECS Dept., University of Michigan, Ann Arbor, MI 48109
Email: mgreyes@umich.edu, neuhoff@umich.edu

***Abstract*— This paper introduces methods for losslessly encoding a Markov random field (MRF) with arithmetic coding. The issues are how to choose the pixel scan order and how to produce coding distributions to accompany the pixels. For an MRF based on an acyclic graph, we choose a scan consistent with the graph and use Belief Propagation (BP) to efficiently compute the optimal coding distributions. For an MRF based on a cyclic graph, we use Local Conditioning (LC) to losslessly encode an appropriately chosen scan of a loop cutset, whose removal leaves an acyclic graph whose pixels can be encoded by the previous method. The results include BP-like formulas for LC in an undirected graph and a formula for the complexity of LC in a cyclic graph. As a first application of the methods, preliminary results of applying the method to an Ising model are given.**

I. Introduction

Markov random fields (MRFs) are often proposed as natural models for spatially distributed data such as images [1], [4], [13]. Their popularity and relevance derives in part from the natural assumption that for such data, the information in some region can be estimated optimally from the information in some sufficiently thick surrounding region. They have been effectively used in problems of segmentation [8], inference [5], [9], restoration [1], [4], computer vision [13], and modeling [7]. Despite their frequent use in such problems, there has been little development of data compression algorithms for MRFs. Indeed the only work of which the authors are aware are the lossy image compression methods of [6] and [10], [11].

The present paper introduces approaches to applying arithmetic coding (AC) to do optimal or nearly optimal lossless encoding of an image modeled as an MRF with known parameters. One motivation is the competitive lossy coding scheme for bilevel images of [10] in which the rows and columns of a square grid of lines from the image are losslessly encoded. At the decoder, the interiors of the blocks formed by the grid are interpolated from their boundaries using an MRF model. In [10], AC was applied without taking into account the MRF model, so the lossless coding was suboptimal. The principal outcomes of this paper are efficient algorithms for computing optimal coding distributions for MRFs on acyclic graphs and optimal context-based coding distributions for MRFs on cyclic graphs. For the cyclic case, we extend the theory of Local Conditioning (LC) to undirected graphs and state new results for belief formulas and complexity.

The *direct approach* to applying AC to an image, is to first *scan* the pixels, i.e., to arrange them into a one-dimensional sequence, $\mathbf{x} = (x_1, x_2, \dots, x_N)$. Then for $i = 1, \dots, N$, the ith pixel x_i is fed to the arithmetic *encoder* along with a *coding distribution* f_i, which is a function $f_i : \mathcal{X} \to [0, 1]$, $\sum_{x \in \mathcal{X}} f_i(x) = 1$, where $\mathcal{X}$ is the finite set of possible pixel values. Ordinarily, f_i will also depend on some or all of the previous pixels $x_1^{i-1} = (x_1, \dots, x_{i-1})$, but this is not reflected in the notation. The encoder outputs a sequence of bits from which the arithmetic *decoder* recreates $\mathbf{x}$. After using a prefix of the encoded bits to decode x_1^{i-1}, it uses subsequent bits and f_i (which it can compute since $x_1, \dots, x_{i-1}$ are known) in the decoding of x_i. For the purposes of this paper, it is not necessary to know the details of AC encoding and decoding. Instead, it suffices to state the well known fact [12] that the number of bits produced by AC when encoding $\mathbf{x}$ with the sequence of coding distributions $f_1, \dots, f_N$ is

$$l(\mathbf{x}) \approx -\log \left(\prod_{i=1}^{N} f_i(x_i) \right) = -\sum_{i=1}^{N} \log f_i(x_i) ,$$

where all logs have base 2. Thus, when applying AC to an image, the main issues to be addressed are the choices of scan order and coding distributions $f_1, \dots, f_N$. The goals are to make $l(\mathbf{x})$ small (on the average) and for the f_i's to be computable with few operations and little storage.

When the image is modeled as a random field $\mathbf{X} = X_1^N$ and when for each i we choose

$$f_i(x) = \Pr(X_i = x | X_{C_i} = x_{C_i}) \triangleq p_{i|C_i}(x_i | x_1^{i-1}) ,$$

($C_i \subset \{1, \dots, i-1\}$ is the context), it follows easily that

$$\mathbb{E}\, l(\mathbf{X}) \approx \sum_{i=1}^{N} \mathbb{E}[-\log f_i(X_i)] = \sum_{i=1}^{N} H(X_i | X_{C_i}) ,$$

where $\mathbb{E}$ denotes expected value and H denotes entropy. If $C_i = \{1, \cdots, i-1\}$, the righthand side of the above equals the entropy $H(\mathbf{X})$ of the random field which is the smallest (average) number of bits into which $\mathbf{X}$ can be encoded. For this reason, the coding distributions are denoted $p_{i|*}(x_i)$ and called *optimal*; the encoding is called *optimal direct encoding*. If $C_i \subseteq \{1, \dots, i-1\}$ where each C_i has size k, the encoding is called *optimal direct* (*k-th order*) *context-based encoding*.

When the MRF is defined on an acyclic graph, optimal direct encoding can be performed in two stages. In the first, the graph is viewed as a tree with the first scanned pixel as its root. The Belief Propagation (BP) algorithm is run, starting at the leaves, with *messages* being passed from children to parents, until messages are received at the root. These messages do not depend on the pixel values. In the second stage, for $i = 1, \dots, N$, $p_{i|*}$ is computed from x_1^{i-1} and the previously computed messages. It is presented to the encoder along with

978-1-4244-4313-0/09/$25.00 ©2009 IEEE

x_i. Likewise, the decoder operates in two stages. It first computes the messages. Then for $i = 1, \ldots, N$, after decoding x_1^{i-1}, it uses them and the previously computed messages to determine $p_{i|C_i}$, which it uses in the decoding of x_i. Overall, the complexity is proportional to N.

In an MRF defined on a cyclic graph, the cycles prevent efficient computation of the $p_{i|*}$. In this case we proceed by partitioning the pixels into two groups. The first is called a *loop cutset* L, in that removing its members from the graph eliminates, i.e. cuts, all loops [9]. That is, it induces an acyclic graph. To encode the image, we first apply direct coding to the loop cutset. Next, we encode the remaining pixels conditioned on the values of the pixels in the loop cutset, using the efficient BP-based method for acyclic graphs described above.

To encode the loop cutset we order its nodes as $l_1, \cdots, l_{|L|}$ in a way that for each l_i, C_{l_i} contains at least one of the closest neighbors of pixel l_i, and compute the coding distribution $p_{l_i|C_{l_i}}$. Local Conditioning is a variant of BP that allows us to do this efficiently in cyclic graphs. Originally, Global Conditioning was introduced by Pearl [9] as a means of applying BP to graphs with cycles. It involves successively fixing the values of the loop cutset nodes, running BP, then combining the results in the appropriate manner. In [2], LC was presented as a modification that enabled considerable computational savings over Pearl's conditioning algorithm and an analytical justification for LC was given in [3] in the case of directed graphs. Because of the directed edges, the analysis was local in the sense that it considered the complexity of sending messages over a particular edge. In Section V we discuss LC and state new formulas for the beliefs and messages as well as the complexity of LC in undirected graphs. Even with LC, in general it will not be possible to directly optimally encode the loop cutset. Thus for a loop cutset node l_i, the context C_{l_i} in general will be a strict subset of $\{l_1, \ldots, l_{i-1}\}$.

In the rest of this paper, Sec. II provides background on MRFs and BP, Sec. III gives the algorithm to compute the optimal direct coding distributions of an acyclic MRF, Sec. IV gives the algorithm to compute the optimal k-th order coding distributions of a cyclic MRF, and Sec. V discusses LC and state new results on the update equations and complexity of LC in undirected graphs. Finally, Sec. VI applies these ideas to an Ising model defined on a grid graph.

II. Background: Markov random field image models and belief propagation

We consider a Markov random field on a graph $G = (V, E)$, where V is a set of N *nodes* (also called *pixels*), and E is a set of undirected *edges*, each connecting a pair of elements of V. For any $U \subset V$, its *neighborhood* ∂U is the set of nodes not in U connected by an edge to a member of U. As a shorthand, ∂i denotes $\partial\{i\}$, $i \in V$.

A Markov random field $\mathbf{X} = \{X_i : i \in V\}$ with finite alphabet $\mathcal{X}$ on the graph $G = (V, E)$ is determined as follows. For each node $i \in V$, X_i is a random variable taking values in $\mathcal{X}$. For a subset of nodes $U \subset V$, $X_U = \{X_i : i \in U\}$ is the subfield on U taking values in $\mathcal{X}_U$. For each edge $\{i, j\} \in E$ there is a nonnegative function $\Psi_{i,j} : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}_+$ called an *edge potential*[1]; and for each node $i \in V$, there is a nonnegative function $\Phi_i : \mathcal{X} \longrightarrow \mathbb{R}_+$ called a *self-potential*[1]. Then for any $\mathbf{x} = \{x_i : i \in V\}$, the probability that $\mathbf{X}$ equals $\mathbf{x}$ is given by the *Gibbs distribution*

$$p(\mathbf{x}) \ = \ \frac{1}{Z} \prod_{\{i,j\} \in E} \Psi_{i,j}(x_i, x_j) \prod_{i \in V} \Phi_i(x_i), \qquad (1)$$

where

$$Z \triangleq \sum_{\mathbf{x}} \prod_{\{i,j\} \in E} \Psi_{i,j}(x_i, x_j) \prod_{i \in V} \Phi_i(x_i)$$

is called the *partition function*. For a subset of nodes $A \subset V$, we let $Z_A(x_A)$[2] denote the partial sum over all configurations $\mathbf{x}'$ where $x'_A = x_A$, from which the partition function can be expressed as $Z = \sum_{x_A} Z_A(x_A)$. Because the probability that X_A equals x_A is proportional to $Z_A(x_A)$, the vector $Z_A = (Z_A(x_A) : x_A \in \mathcal{X}_A)$ is called the *belief* for A. It follows from (1) that $\mathbf{X}$ is *Markov* with respect to G in the sense that for each i, random variable X_i is conditionally independent of all others in $\mathbf{X}$ given the values of its neighbors [7]. Depending on whether the underlying graph for an MRF is cyclic or acyclic, we say that we have a *cyclic* or *acyclic* MRF, respectively.

Given an image $\mathbf{x}$ and an ordering $x_1, \cdots, x_N$ of the pixels, one can compute the conditional probability $p_{i|C_i}(x_i \mid x_{C_i})$ of the ith pixel value given the values on its context by

$$p_{i|C_i}(x_i \mid x_{C_i}) = \frac{Z_{i \cup C_i}(x_i, x_{C_i})}{Z_{C_i}(x_{C_i})}. \qquad (2)$$

In general, computing Z_A for a given subset of nodes is nontrivial, as it involves a summation over a number of terms that is exponential in $N - |A|$. However, in the case that the underlying graph G is acyclic, there is an efficient algorithm, called *Belief Propagation* (BP) [9], for computing various beliefs. The algorithm involves a sequence of recursive message passing operations between neighboring nodes, where a message is a table of numbers each corresponding to a value in the alphabet of the receiving node.

The recursion begins with each leaf node j sending to its unique neighbor δj the message component

$$m_{j \to \delta j}(x_{\delta j}) \ = \ \sum_{x_j} \Phi_j(x_j) \Psi_{j,\delta j}(x_j, x_{\delta j}) \qquad (3)$$

for each $x_{\delta j} \in \mathcal{X}$. The algorithm proceeds with each non-leaf node j passing to neighbor $i \in \delta j$ the message $m_{j \to i}$ with components determined by

$$m_{j \to i}(x_i) = \sum_{x_j} \Psi_{j,i}(x_j, x_i) \Phi_j(x_j) \prod_{k \in \partial j \setminus i} m_{k \to j}(x_j) \ ,$$

with the rule that node j does not form the message to send to i until receiving incoming messages from all other neighbors. The messages $\{m_{j \to i}, m_{i \to j} : \{i, j\} \in E\}$ can be used to

[1]Actually, it is more common to call $-\log \Psi$ and $-\log \Phi$ *potentials*.

[2]We will use the shorthand Z_i for $Z_{\{i\}}$

compute, for example, the belief for a connected subset of nodes $A \subseteq V$ [5]:

$$Z_A(x_A) = \prod_{a \in A} \Phi_a(x_a) \prod_{\{k,l\} \subset A} \Psi_{k,l}(x_k, x_l) \prod_{j \in \partial A} m_{j \to i}(x_i) \ . \tag{4}$$

where the second product is restricted to $\{k, l\} \in E$, and i is the unique neighbor of j in A.

We can also express the belief propagation equations in matrix notation. For edge $\{i, j\}$, we define $A_{i,j} = [\Psi_{i,j}(x_i, x_j)]$ be the matrix of potentials between different values of x_i, x_j, where the rows correspond to values of x_j and columns values of x_i. If we let $m_{j \to i}$ denote the vector $[m_{j\to i}(0), \ldots, m_{j\to i}(|\mathcal{X}| - 1)]^T$, Φ_j the vector $[\Phi_j(0), \ldots, \Phi_j(|\mathcal{X}|)]^T$, $\odot$ component-wise multiplication, and $\cdot$ ordinary matrix multiplication, the message recursion is

$$m_{j \to i} = A_{j,i} \cdot \left[\Phi_j \odot \prod_{k \in \partial j \setminus i}^{\odot} m_{k \to j} \right] , \tag{5}$$

while the belief at node i is

$$Z_i = \Phi_i \odot \prod_{k \in \partial i}^{\odot} m_{k \to i}. \tag{6}$$

The complexity of computing a single message $m_{j \to i}$ is proportional to $|\mathcal{X}|^2$ and two messages must be computed per edge. In an acyclic graph the number of edges is proportional to N, and BP has complexity linear in N.

III. Encoding an acyclic MRF

To encode the pixels we order them into a scan as $1, 2, \ldots, N$. If the scan has the property that when the acyclic graph G is viewed as a tree rooted at the first scanned node 1, each subsequent node i is a child of some previously scanned node $\pi(i)$, we say that the scan is *connected*. The following proposition, which can be proved using (4), shows how the $p_{i|*}$'s can be efficiently computed with a connected scan.

Proposition 3.1: Let $\mathbf{x} = (x_1, \cdots, x_N)$ be an image from an acyclic MRF. If the scan is connected, then

$$p_{i|C_i}(x_i) = \frac{\Psi_{\pi(i),i}(x_{\pi(i)}, x_i)\Phi_i(x_i) \prod\limits_{j \in \partial i \setminus \pi(i)} m_{j \to i}(x_i)}{\sum\limits_{x_i'} \Psi_{\pi(i),i}(x_{\pi(i)}, x_i')\Phi_i(x_i') \prod\limits_{j \in \partial i \setminus \pi(i)} m_{j \to i}(x_i')} \ . \tag{7}$$

With this proposition in mind, one can efficiently encode an image $\mathbf{x}$ with a two-stage process. The first pixel to be scanned is chosen arbitrarily. The graph is viewed as a tree with this pixel as its root. The scan order for the remaining pixels is arbitrary as long as it is connected. In the first stage, starting at the leaves, BP successively computes and passes messages from children to parents, using (3) and (4). In the second stage, for $i = 1$ to N, (7) is used to compute $p_{i|C_i}(\cdot)$ from x_1^{i-1} and the previously computed messages. One feeds it along with x_i to the AC encoder.

In Summary, if the MRF is acyclic and the scan is connected, we can reuse the BP messages to compute the optimal direct coding distributions. The complexity of computing the optimal coding distributions in this manner is $\mathcal{O}(N)$.

IV. Encoding a cyclic MRF

For an MRF $\mathbf{X}$ whose graph G is cyclic, the approach described above is infeasible, as the cycles prevent BP from being used to compute the optimal coding distributions. To circumvent this, we consider an approach in which one directly encodes the pixels X_L corresponding to a set of nodes L, called a *loop cutset*, whose removal leaves an acyclic graph. Since the latter is acyclic, the BP method of the previous section can be used to optimally encode $X_{V \setminus L}$ conditioned on the previously encoded values of X_L. Thus our algorithm for an acylic MRF has two phases – encoding the loop cutset, and conditionally encoding the acyclic remainder. We discuss these in the following subsections in reverse order.

A. Optimal conditional encoding

If L is a loop cutset of G, then conditioning on $X_L = x_L$ effectively removes all cycles from G [9], permitting the use of BP to compute the conditional probabilities required to directly and optimally encode $x_{V \setminus L}$ conditioned on x_L. To see this, it is helpful to visualize the *unwrapped graph* T_L as defined below. We note, however, that T_L is simply an aid to understanding how BP can be used exactly in a cyclic graph after conditioning on the values of a loop cutset L.

Given a cyclic graph $G = (V, E)$ and a loop cutset $L \subset V$, we create the unwrapped graph $T_L = (\bar{V}, \bar{E})$ by first discarding each loop cutset node $j \in L$ and all edges incident to j. By definition, this removes all cycles. Then, to each node $i \in V \setminus L$ that was a neighbor of some $j \in L$, we attach a *copy* $j^{(i)}$ of node j. Then $\bar{V} = (V \setminus L) \cup \bar{L}$, where $\bar{L} = \{j^{(i)} : j \in L, i \in \partial j\}$ is the set of copy nodes. The edge set $\bar{E}$ consists of edges in E not incident to nodes in L, plus "reconnected" edges of the form $\{i, j^{(i)}\}$ for $j \in L$, $i \in \partial j$. An unwrapped graph is illustrated in Figure 1(b). In a way that causes the messages computed on T_L to give the correct conditional probabilities we assign potentials to the nodes and edges of T_L. Specifically, for $i, j \in V \setminus L$, $\bar{\Psi}_{i,j} \equiv \Psi_{i,j}$ and $\bar{\Phi}_i \equiv \Phi_i$; for a copy $j^{(i)}$, $i \in \partial j$, of loop cutset node j, $\bar{\Psi}_{i,j^{(i)}} \equiv \Psi_{i,j}$ and $\bar{\Phi}_{j^{(i)}} \equiv \Phi_j^{\delta}$, where $\delta = |\partial j|^{-1}$ and $|\partial j|$ is the number of neighbors of j. This defines an MRF $\bar{\mathbf{X}}$ on T_L such that $\bar{x}_{j^{(i)}}$ is fixed to the observed value x_j for all $j^{(i)}$, $j \in L, i \in \partial j$.

We now order the pixels in $V \setminus L$ for encoding, say $i_1, \ldots, i_{|V \setminus L|}$, subject to the constraint that for each k, i_k is a child of a pixel in $\{i_1, \ldots, i_{k-1}\}$. We run BP on T_L and use (7) to compute the conditional probabilities $\{p_{i|*}^{(L)}\}$, where the superscript (L) indicates the implicit conditioning on x_L. In running BP, we note that (copies of) the cutset nodes L are leaves in T_L, but unlike the usual leaves, the corresponding pixel values are fixed. As a result, for $j^{(i)}$, $j \in L, i \in \partial j$, the sum in (3) reduces to just one term.

In summary, applying the BP-based encoding described in Section III to $X'_{V\setminus L}$ results in a low complexity method of optimal conditional encoding, producing $H(X_{V\setminus L}|X_L)$ bits on the average. In particular, its complexity is proportional to the number of nodes/edges in the unwrapped tree T_L, which is ordinarily proportional to N.

B. Encoding a loop cutset

To encode the pixels of the loop cutset L, we order them as $l_1, l_2, \cdots, l_{|L|}$. As mentioned earlier and explained in the next section, we use Local Conditioning (LC) to compute coding distributions $\{p_{l_i|C_{l_i}}\}$. As we will see, for each l_i there is a subset $R_{l_i} \subset L$ of loop cutset nodes such that after running LC, node l_i can compute the belief $Z_{l_i \cup R_{l_i}}$. If the context C_{l_i} is contained in R_{l_i}, then l_i can compute the beliefs $Z_{l_i \cup C_{l_i}}$ and $Z_{C_{l_i}}$, from which the coding distributions $p_{l_i|C_{l_i}}$ can be computed. For each l_i, we want to choose C_{l_i} so that it contains those loop cutset nodes in R_{l_i} that are closest to l_i and precede l_i in the scan.

V. Local Conditioning in Undirected Graphs

In this section we briefly discuss Global Conditioning (GC) and LC in undirected graphs. The following extension is new, and the details will be discussed rigorously in a forthcoming paper. Let L be a loop cutset and assume for the moment that the unwrapped T_L is connected. For each instantiation x_L of the loop cutset nodes we define $Z_i^{(x_L)}$ to be the *conditional belief* at node i given x_L. Similarly, we refer to $m_{k\to i}^{(x_L)}$ as the *conditional message* from k to i given x_L. We can simultaneously compute the conditional messages and beliefs for all possible loop cutset configurations by passing large message arrays of the form $M_{k\to i}^{(L)} = [m_{k\to i}^{(x_L)} : x_L \in \mathcal{X}_L]$ where each conditional message is processed separately as in (5). After all message passing is completed, each node i will compute the belief array $Z_i^{(L)} = [Z_i^{(x_L)} : x_L \in \mathcal{X}_L]$, from which the belief Z_{C_i} can be computed as $Z_{C_i} = \sum_{x_{L\setminus C_i}} Z_i^{(x_L)}$, where the summation is over $x_{(L\setminus i)\setminus C_i}$ in the case that $i \in L$. This is the GC algorithm and the complexity is $\mathcal{O}(|\mathcal{X}|^{|L|})$. This is because the complexity of computing a conditional message is proportional to $|\mathcal{X}|^2$, so the complexity of computing a message array $M_{k\to i}$ is proportional to the number of columns of $M_{k\to i}$.

We now describe how we can reduce the complexity of GC using LC. For a loop cutset node $l \in L$, we define $T_L^{(l)}$ to be the subgraph consisting of the union of paths connecting copies of l in T_L. For a given edge $\{i, k\}$, $T_{L,i\setminus k}$ denotes the component of $T_L \setminus \{k\}$[3] containing i. We say that a loop cutset node l is *downstream* of message $M_{k\to i}$ if $l^{(j)} \in T_{L,i\setminus k}$ for all $j \in \partial l$ and $L_{i\setminus k} \subseteq$ is the set of loop cutset nodes downstream of $M_{k\to i}$. The set of loop cutset nodes that are not downstream of $M_{k\to i}$ is $L_{i\setminus k}^c$. If a cutset node l is downstream of message $M_{k\to i}$, then for each configuration x_L, the conditional message $m_{k\to i}^{(x_L)}$ is not a function of x_l.

[3]This is obtained by removing k and all incident edges.

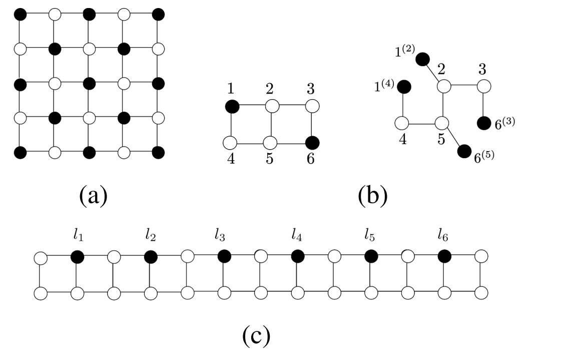

Fig. 1. (a) A 5×5 grid graph with checkerboard loop cutset L_1. (b) A 2×3 grid graph and the unwrapped network T_L corresponding to loop cutset $L = \{1, 6\}$. (c) A 2×13 grid graph. All loop cutset nodes indicated in black.

This is because $m_{k\to i}^{(x_L)}$ is function consisting of sums and products of potentials for nodes and edges contained in the subtree $T_{L,k\setminus i}$ and no copy of a downstream node l is in the subtree $T_{L,k\setminus i}$. This leads to the following lemma.

Lemma 5.1: Let l be a loop cutset node and let x_L and x'_L be configurations such that $x_{L\setminus l} = x'_{L\setminus l}$ and $x_l \neq x'_l$. If l is downstream of $M_{k\to i}$, then

$$m_{k\to i}^{(x_L)} = m_{k\to i}^{(x'_L)}.$$

To eliminate redundancy and reduce computation required for conditioning on x_L, we consider the *reduced messages* $m_{k\to i}^{(x_{L^c_{i\setminus k}})}$ such that $m_{k\to i}^{(x_{L^c_{i\setminus k}}, x'_{L_{i\setminus k}})} = m_{k\to i}^{(x_{L^c_{i\setminus k}})}$ for all $x'_{L_{i\setminus k}}$. The message array $M_{k\to i}^{L^c_{i\setminus k}}$ has fewer columns than $M_{k\to i}^{(L)}$, and the resulting BP complexity is $\mathcal{O}(|\mathcal{X}|^{|L^c_{i\setminus k}|}$ rather than $\mathcal{O}(|\mathcal{X}|^{|L|})$. However, node i must be able to enlarge the reduced message array in order to combine it with other incoming messages. For subsets $B \subset A$, we let $E_B^A : M_{k\to i}^{(B)} \mapsto M_{k\to i}^{(A)}$ be the operator that *expands* message $M_{k\to i}^{(B)}$ to message $M_{k\to i}^{(A)}$ through the relation $m_{k\to i}^{(x_B, x_{A\setminus B})} = m_{k\to i}^{(x_B)}$ for all $x_B \in \mathcal{X}_B$, $x_{A\setminus B} \in \mathcal{X}_{A\setminus B}$. The operation $E_{L^c_{i\setminus k}}^L(\cdot)$ 'expands' the matrix $M_{k\to i}^{(L^c_{i\setminus k})}$ to the matrix $M_{k\to i}^{(L)}$ to be used to compute the belief $Z_i^{(L)}$ or an outgoing message $M_{i\to j}^{(L)}$, for some neighbor $j \in \partial i \setminus k$.

A loop cutset node l is said to be *upstream* of message $M_{k\to i}$ if $l^{(j)} \in T_{L,k\setminus i}$ for all $j \in \partial l$. Therefore, $L_{k\setminus i}$ is the set of loop cutset nodes that are upstream of $M_{k\to i}$. We define $R_{ik} \subset L$ to be the set of loop cutset nodes that are neither upstream nor downstream of messages $M_{k\to i}$ and $M_{i\to k}$. We define the *summed out message* $\tilde{M}_{k\to i}^{(R_{ki})} = [\tilde{m}_{k\to i}^{(x_{R_{ki}})}]$ determined by

$$\tilde{m}_{k\to i}^{(x_{R_{ki}})} = A_{k,i} \cdot \sum_{x_{L_{k\setminus i}}} \prod_{j\in\partial k\setminus i}^{\odot} m_{j\to k}^{(x_{L_{k\setminus i}}, x_{L'})},$$

where $L' = L^c_{k\setminus j} \setminus L_{k\setminus i}$. For edge $\{i, k\}$ and node i, we say that R_{ik} and $R_i = \bigcup_{k\in\partial j} R_{ki}$ are the respective *relevant* subsets of loop cutset nodes. The following new theorem establishes the LC propagation formulas for undirected graphs.

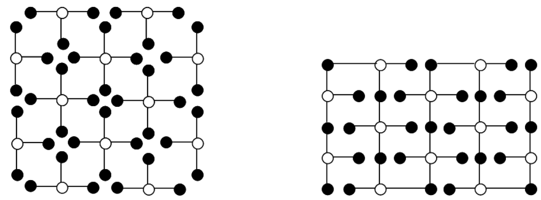

Fig. 2. (a) Unwrapped graph T_{L_1} from 7×7 grid with checkerboard loop cutset L_1. (b) Reidentified tree $\bar{T}_{L_1}$.

Theorem 5.2: The LC formulas (in matrix notation) are

$$Z_i^{(R_i)} = E_i^{R_i}(\Phi_i) \odot \prod_{k \in \partial i}^{\odot} E_{R_{ki}}^{R_i}(\tilde{M}_{k \to i}^{(R_{ki})}).$$

and

$$\tilde{M}_{i \to k}^{(R_{ik})} = A_{i,k} \cdot \left[E_i^{R_i}(\Phi_i) \odot \prod_{j \in \partial i \setminus k}^{\odot} E_{R_{ji}}^{R_i}(\tilde{M}_{j \to i}^{(R_{ji})}) \right] \cdot s_i^k,$$

where s_i^k is a $|R_i| \times |R_{ik}|$ matrix that performs the 'summing out' over the loop cutset nodes in $\cup_{j \in \partial i \setminus k} R_{ji}$.

Since the message $\tilde{M}_{k \to i}$ is conditioned only on the l's in R_{ki}, the complexity of LC is dominated by the maximum of $|R_{ki}|$ over all edges. We can now state our result for the complexity of LC in undirected graphs.

Theorem 5.3: The complexity of LC using loop cutset L is $\mathcal{O}(|E||\mathcal{X}|^{c(L)})$, where we say the *cost* of L is

$$c(L) = \max_{\{i,j\} \in E} |R_{ij}|. \quad (8)$$

Once node $l \in L$ computes $Z_l^{(X_{R_l})}$ from its incoming messages, it can compute $Z_{l \cup B}(x_l, x_B)$ for $B \subset R_l$. Therefore, if $C_l \subset R_l$, the optimal context-based coding distributions can be computed. If $C_l \not\subseteq R_l$, additional conditioning will have to be used. To optimize computational efficiency, we can optimize the quantity $c(L)$ over all loop cutsets. If the unwrapped graph T_L is a forest, we can form an unwrapped graph T'_L as illustrated, for example, in Figure 2 (a) and (b). In this case we refer to the cost $c(T'_L)$ of the unwrapped graph and new message processing rules are needed at the loop cutset nodes. Nevertheless we can still encode the loop cutset pixels efficiently.

VI. Example: Grid Graphs

We now consider grid graphs $G = (V, E)$, where V is an $m \times n$ portion of the positive integer lattice and E is the set of horizontally and vertically adjacent neighbors. Since coding $V \setminus L$ conditioned on L is optimal and efficient, we focus this example on coding the latter. Let L_1 be the checkerboard loop cutset and $\bar{T}_{L_1}$ the reidentified tree in Figure 2 (a) and (b), respectively. The following theorem is straightforward.

Theorem 6.1: Let L_1 be the checkerboard loop cutset of and $m \times n$ grid graph and let $\bar{T}_{L_1}$ be the unwrapped graph described above. The complexity of encoding L with LC on $\bar{T}_{L_1}$ is

$$\mathcal{O}(mn|\mathcal{X}|^{\min\{m,n\}}).$$

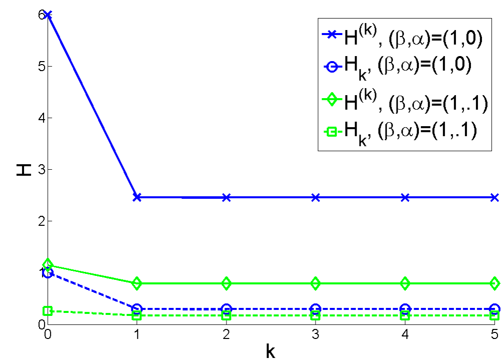

Fig. 3. Entropy of kth order coding of the MRF showns in Fig. 1 (c).

Now let us consider k-th order context coding. To see how large k needs to be, we have so far considered an Ising model defined on the 2×13 grid graph shown in Fig. 1. Each X_i takes values in $\{-1, +1\}$ where $\Psi_{i,j}(x_i, x_j) = \exp\{\beta x_i x_j\}$ and $\Phi_i(x_i) = \exp\{\alpha x_i\}$ for $\alpha, \beta \geq 0$. To see how AC will perform, we use LC to compute $H_{(k)} \triangleq H(X_{l_6} | X_{l_{6-k}}, \ldots, X_5)$ and $H^{(k)} \triangleq \sum_{i=1}^{6} H(X_{l_i} | X_{l_{i-k}}, \ldots, X_{l_{i-1}})$, for $k = 0, \ldots, 6$ and two different choices of α, β. Since both $H_{(k)}$ and $H^{(k)}$ plateau at $k = 1$, we discover the interesting fact that, at least in this example, there are basically no gains to choosing $k > 1$.

VII. Conclusion

In this paper we have introduced the problem of arithmetic encoding an MRF. We have presented algorithms for encoding acyclic or cyclic MRFs based on BP messages and our extension of the theory of LC to undirected graphs.

References

[1] J. Besag, "On the Statistical Analysis of Dirty Pictures", *J. of Roy. Stat. Soc. B*, vol. 48, no. 3, pp. 256-302, 1986.
[2] F.J. Diez, "Local conditioning in Bayesian networks", *Artificial Intelligence*, 87, pp. 1-20, 1996.
[3] A. Fay and J-Y Jarrray, "A Justification of Local Conditioning in Bayesian Networks", *Int. Jrnl. Apprx. Reasoning*, vol. 24, pp. 59-81.
[4] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE Trans. PAMI*, vol. 6, pp 721-741, Nov. 1984.
[5] F. Jensen, *An Introduction to Bayesian Networks*, UCL Press, London,1996.
[6] I. Kontoyiannis, "Pattern Matching and Lossy Data Compression on Random Fields", *IEEE Trans. on Info. Theory*, vol. 49, no. 4, April 2003.
[7] S. Lauritzen, *Graphical Models*, Oxford University Press, 1996.
[8] T.N. Pappas, "An adaptive clustering algorithm for image segmentation", *IEEE Trans. Signal Process.*, 40 (1992), pp. 901913.
[9] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Francisco, CA, 1988.
[10] M. G. Reyes, X. Zhao, D. L. Neuhoff, T. N. Pappas, "Lossy Compression of Bilevel Images Based on Markov Random Fields", ICIP, Sept. 2007.
[11] M. G. Reyes, X. Zhao, D. L. Neuhoff, T. N. Pappas, "Structure-preserving properties of bilevel image compression", HVEI XII, San Jose, CA, January 2008.
[12] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann, San Francisco, 2006.
[13] G. Winkler, *Image Analysis, Random Fields and Dynamic Monte Carko Methods: A Mathematical Introduction*. Berlin: Springer-Verlag, 1995.