Lossy Source Compression Using Low-Density Generator Matrix Codes: Analysis and Algorithms

Martin J. Wainwright, Elitza Maneva, and Emin Martinian

***Abstract*—We study the use of low-density generator matrix (LDGM) codes for lossy compression of the Bernoulli symmetric source. First, we establish rigorous upper bounds on the average distortion achieved by check-regular ensemble of LDGM codes under optimal minimum distance source encoding. These bounds establish that the average distortion using such bounded degree families rapidly approaches the Shannon limit as the degrees are increased. Second, we propose a family of message-passing algorithms, ranging from the standard belief propagation algorithm at one extreme to a variant of survey propagation algorithm at the other. When combined with a decimation subroutine and applied to LDGM codes with suitably irregular degree distributions, we show that such a message-passing/decimation algorithm yields distortion very close to the Shannon rate-distortion bound for the binary symmetric source.**

***Index Terms*—Lossy source coding, graphical codes, low-density generator matrix (LDGM) codes, satisfiability problems, MAX-XORSAT, message-passing, belief propagation, sum-product, survey propagation.**

I. Introduction

A LARGE class of codes, including turbo codes [3] and low-density parity check (LDPC) codes [17], are most naturally represented as factor graphs [44], [23]. When the degrees are bounded independently of blocklength, these graphs are known to be "locally treelike", meaning that the length of the typical cycle scales logarithmically in the blocklength [42]. This structure makes such codes ideally suited to approximate decoding using algorithms and relaxations known to be exact on trees, such as the sum-sinproduct algorithm [23], [25], or tree-based linear programming relaxations [13]. Indeed, the past decade has witnessed a flurry of research, and current understanding of such graphical codes and message-passing algorithms for channel coding is now relatively mature [42], [26], [41], [5]. A related line of research [18], [43] has shown that good channel codes, such LDPC codes and turbo codes, can be leveraged for problems of *lossless* compression, including the Slepian-Wolf problem. In contrast, for problems of *lossy source coding*, there remain a variety of open questions and issues associated with the use of sparse graph codes and associated message-passing algorithms.

The idea of using linear codes for lossy compression of discrete sources is a classical one. The dissertation work of Goblick [19] shows that the rate-distortion bound for binary sources under Hamming distortion can be achieved using linear codes. Viterbi and Omura [45] derive bounds on the distortion achievable using trellis codes, applicable to memoryless sources and bounded distortion measures, and show that the obtained distortion approaches the Shannon limit as the trellis constraint length increases. This trellis-based approach was extended by forming the duals of trellis-coded modulation schemes, yielding the approach of trellis-coded quantization [28]. The advantage of all these trellis constructions is that exact encoding and decoding can be performed using the max-product or Viterbi algorithm [25], with complexity that grows linearly in the trellis length but exponentially in the constraint length. However, saturating rate-distortion bounds requires increasing the trellis constraint length [45], which incurs exponential complexity (even for the max-product or sum-product message-passing algorithms). Other work [33] shows that it is possible to approach the binary rate-distortion bound using LDPC-like codes, albeit with degrees that grow logarithmically with the blocklength.

The focus of this paper is the use of low-density generator matrix (LDGM) codes for lossy compression problems. As the dual of an LDPC code, any LDGM code shares a similar representation in terms of a sparse factor graph, making it amenable to efficient message-passing algorithms. There is now an on-going line of work on the use of LDGM codes for lossy data compression. Martinian and Yedidia [32] studied the binary erasure quantization, a special compression problem dual to binary erasure channel coding, and showed that LDGM codes combined with modified message-passing can saturate the associated rate-distortion bound. In other work, various researchers [6], [7], [39], [38] have used heuristic techniques from statistical physics, including the cavity and replica methods, to provide non-rigorous analyses of LDGM rate-distortion performance for the binary symmetric source. In the special case of zero-distortion, these calculations have been made rigorous by several independent groups [10], [35], [8], [12]. Other researchers have proposed message-passing schemes,

Manuscript received August 22, 2007; revised April 28, 2009. Current version published March 10, 2010. The work of M. J. Wainwright was supported by the National Science Foundation under Grants CAREER-CCF-0545862 and DMS-0528488, an Alfred P. Sloan Foundation Fellowship, and a grant from the Microsoft Corporation. The work of E. Maneva was supported in part by the National Science Foundation under Grant DMS-0528488. The material in this paper was presented in part at International Symposium on Information Theory Adelaide, Australia, September 2005.

M. J. Wainwright is with the Department of Statistics, and Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 USA.

E. Maneva is with the Institut de Matematica, Universitat de Barcelona (UB), 08007 Barcelona, Spain.

E. Martinian is with the Tilda Consulting, Boston, MA USA.

Communicated by W. Szpankowski, Associate Editor for Source Coding.

Color versions of Figures 1–3, 8, and 9 in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2009.2039160

0018-9448/$26.00 © 2010 IEEE

including modified forms of belief propagation [38] and variants of survey propagation [6], [7], for performing lossy data compression using either linear codes or non-linear variants.

This paper makes two distinct but complementary contributions to the current understanding of LDGM codes for lossy compression. The first contribution is of a theoretical nature: we provide a rigorous analysis of the effective rate-distortion performance of random ensembles of LDGM codes for the Bernoulli symmetric source. These analytical results establish that LDGM rate-distortion performance under optimal encoding rapidly approaches the Shannon limit as the degrees are increased, and thus validate the use of these codes for lossy compression. Although the current paper focuses purely on the symmetric Bernoulli source, our techniques—namely, analysis over random ensembles using moment methods—have broader applicability to lossy compression of more general discrete and continuous sources. Our second contribution is of a practical nature: we develop a family of message-passing algorithms for computationally efficient source encoding, including both a form of survey propagation [36], [34] at one extreme and standard belief propagation [40], [23] at the other extreme. As with application of survey propagation to solving satisfiability problems, we combine these message-passing updates with a sequence of decimation steps, in which subsets of variables are set and the problem is reduced. By using LDGM codes with suitably designed degree distributions, we obtain practical results extremely close to the Shannon rate-distortion function for the symmetric Bernoulli source; to the best of our knowledge, these results are the best reported thus far in the literature. Since portions of this research were first published in conference form [46], [29], other work has followed up and extended some of our results. Martinian and Wainwright [30], [31], [47] have studied the use of compound constructions, in which an LDGM code is concatenated with an LDPC code, for both lossy compression and binning. Gupta and Verdu [21] have used similar analysis techniques (moment methods) to study the performance of a novel non-linear scheme for lossy compression. Using the algorithms proposed here, Fridrich and Filler [16] have achieved state-of-the-art results in binary steganography; subsequent work by Filler [14] has studied the effects of algorithmic parameters on the rate-distortion performance, and used density-evolution-based methods to compare LDGM degree distributions. Whereas this paper provides achievability results for lossy compression via LDGM codes, other researchers have derived lower bounds on the distortion performance of LDGM codes. These inachievability results include lower bounds by Dimakis *et al.* [11], [48] for random ensembles, and by Kudekar and Urbanke [24] for specific codes.

The remainder of this paper is organized as follows. Section II contains background material on lossy source coding and low-density generator matrix codes. In Section III, we state and then prove our theoretical results on the effective rate-distortion performance of the class of LDGM codes. In Section IV, we describe a class of message-passing algorithms for performing source encoding, and illustrate their practical rate-distortion performance. We conclude the paper with a discussion and future directions in Section V.

II. Background

In this section, we provide relevant background material on the problem of lossy source coding, and low-density generator matrix codes.

A. Lossy Coding of Bernoulli Sources

In this paper, we consider the problem of compressing the *symmetric Bernoulli source*, in which the source sequence $S = (S_1, \ldots, S_n)$ consists of independent and identically distributed (i.i.d.) fair coin tosses (i.e., each S_i is Bernoulli with parameter $\frac{1}{2}$). Effective coding techniques for solving this binary compression problem, though a very special case, serve as a building block for tackling compression of more complex sources. The problem of lossy compression is to represent each source sequence $S \in \{0,1\}^n$ by some codeword in a code $\mathbb{C}$ of size much smaller than the total number of strings—that is, with $|\mathbb{C}| \ll 2^n$. More concretely, one can achieve a given compression rate $R = \frac{m}{n}$ by mapping each n-bit source sequence to a binary string $z \in \{0,1\}^m$. This stored sequence z then defines a reconstructed source sequence $\widehat{S}(z)$, where the source decoding map $z \mapsto \widehat{S}(z)$ depends on the nature of the code construction. (For instance, when using low-density generator matrix codes as in the sequel, this source decoding map is simply matrix multiplication). The quality of the reconstruction is measured in terms of some distortion metric $d : \{0,1\} \times \{0,1\} \rightarrow \mathbb{R}_+$. The natural metric for the symmetric Bernoulli source is *Hamming distortion*, meaning that the distortion is given by $d(\widehat{S}(z), S) = \sum_{i=1}^n |S_i - \widehat{S}_i(z)|$. The optimal source encoder seeks to minimize this distortion, and so searches the codebook to find the minimum distance encoding—namely, $\widehat{z} := \arg\min_z d(\widehat{S}(z), S)$. Classical rate-distortion theory [9] specifies the optimal trade-offs between the compression rate R and the best achievable average distortion $D = \mathbb{E}[d(\widehat{S}, S)]$, where the expectation is taken over the random source sequences S. For the symmetric Bernoulli source with Hamming distortion, it is well known [9] that the rate-distortion function is given by

$$R(D) = \begin{cases} 1 - H(D), & \text{if } D \in [0, 0.5) \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

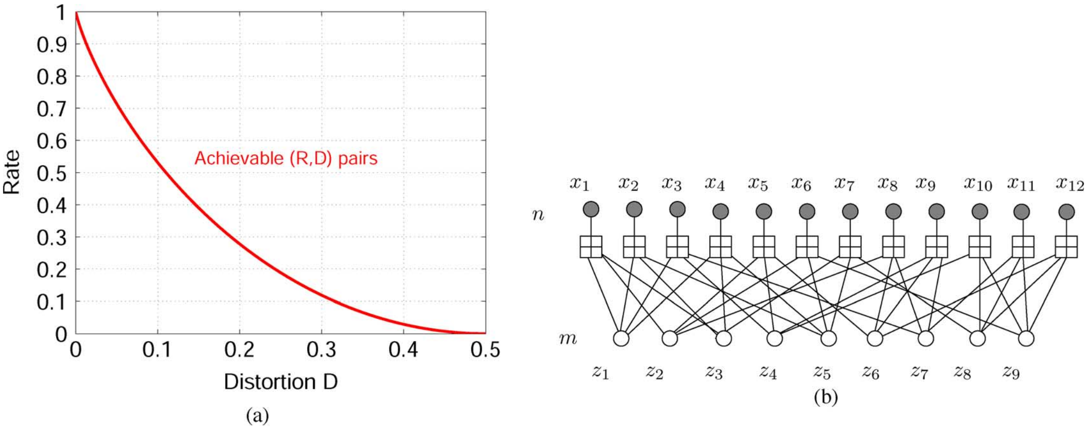
Here $H : [0,1] \rightarrow \mathbb{R}_+$ is the binary entropy function, defined as

$$H(t) := -t\log(t) - (1-t)\log(1-t) \tag{2}$$

for all $t \in (0,1)$, and $H(0) = H(1) = 0$ by continuous extension. See Fig. 1(a) for a plot of the rate-distortion function (1).

B. Low-Density Generator Matrix (LDGM) Codes

A binary linear code $\mathbb{C}$ of rate $R = \frac{m}{n}$ corresponds to an m-dimensional linear subspace of the Boolean hypercube $\{0,1\}^n$, and can be represented by a *generator matrix*, say $G \in \{0,1\}^{n \times m}$. In this generator representation, each codeword $x \in \mathbb{C}$ belongs to the range space of G, and so can be written in the form $x = Gz$, for a suitable sequence of *information bits* $z \in \{0,1\}^m$. Here all arithmetic (addition and multiplication) is performed in modulo two. Presuming that


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Rate
Achievable (R,D) pairs
0
0.1
0.2
0.3
0.4
0.5
Distortion D
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12}
n
m
z_1 z_2 z_3 z_4 z_5 z_6 z_7 z_8 z_9

(a)

(b)

Fig. 1. (a) Plot of rate-distortion function $R(D) = 1 - H(D)$ for the symmetric Bernoulli source. (b) Factor graph representation of low-density generator matrix (LDGM) code. In the top row, a series of n checks (represented by square nodes) are each connected to a source bit (represented by a gray circular node). The checks also connect to a set of m information bits (represented by white circles) in the bottom row. The illustrated code has $n = 12$ and $m = 9$ for design rate $R = 0.75$, and regular check degree $\gamma_c = 3$ and bit degree $\gamma_v = 4$.

G is full rank, the code $\mathbb{C}$ then consists of 2^m possible n-bit strings, and so has rate $\mathrm{R} = \frac{m}{n}$.

The structure of a given generator matrix G can be captured by its factor graph [23], which is a bipartite graph in which circular nodes represent code bits x_i (or rows of G), and square nodes represent the information bits (or columns of G). For instance, Fig. 1(b) shows a binary linear code of blocklength $n = 12$ and $m = 9$ information bits, represented in factor graph form by its generator matrix $G \in \{0,1\}^{12 \times 9}$, with an overall rate of $R = 3/4$. The degrees of the check and variable nodes in the factor graph are $\gamma_c = 3$ and $\gamma_v = 4$, respectively, so that the associated generator matrix G has three “1”s in each row, and four “1”s in each column. When the generator matrix is sparse in this sense, we refer to the resulting code as a *low-density generator matrix* code (or LDGM code for short).

In this paper, we focus on a commonly studied random ensemble of LDGM codes [10], [35], [8], [12] known as the *check-regular ensemble with degree* γ_c, and denoted by $\mathrm{LDGM}(\gamma_c)$. A random code in this ensemble is formed by having each check $j \in \{1, \ldots, m\}$ choose independently a subset of γ_c distinct information bits to which it then connects with edges. Note that this procedure results in a graph in which every check has degree exactly γ_c, and the variable degrees have a random (asymptotically Poisson) distribution.

III. Upper Bounds on LDGM Rate-Distortion

In this section, we analyze the rate-distortion performance of random ensembles of LDGM codes under optimal encoding, more specifically obtaining upper bounds on the rate-distortion performance of the check-regular ensemble of LDGM codes. Alternatively phrased, this result can be viewed as an upper bound on the average ground state energy of the “p-spin” model in statistical physics [35], or in computer science terminology as an upper bound on the average value of a random MAX-XORSAT problem.

To state the problem more precisely, for any fixed code $\bar{\mathbb{C}}$, the optimal source encoder outputs the codeword that minimizes the Hamming distortion to the (random) source sequence S, a quantity referred to as the minimum distance (MD) codeword

$$\widehat{X} := \arg\min_{x \in \bar{\mathbb{C}}} \|x - S\|_1. \tag{3}$$

The optimal encoder thereby achieves the (rescaled) Hamming distortion

$$d_n(S, \bar{\mathbb{C}}) := \frac{1}{n} \|\widehat{X} - S\|_1. \tag{4}$$

Of interest is the rescaled average distortion $\mathbb{E}[d_n(S, \bar{\mathbb{C}})]$, where the average is taken over the symmetric Bernoulli source sequence S.

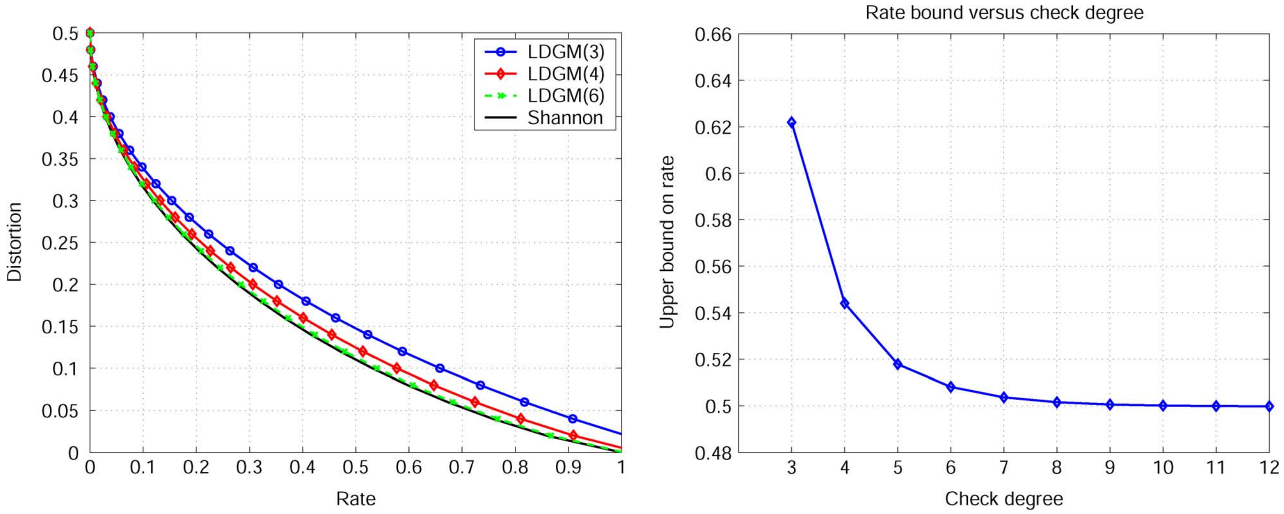
A. Statement and Some Consequences

Our main result provides an upper bound on the minimal rate R required to achieve average distortion D using a code drawn from the check-regular LDGM ensemble. In order to state this result, we require a few definitions. For each $t \in (0, \frac{1}{2}]$ and $D \in [0, \frac{1}{2})$, define the function $F(t; D)$ via

$$\begin{aligned}\inf_{\lambda < 0} \max_{u \in [0, D]} \{ &H(u) - H(D) + u \log[(1-t)\exp(\lambda) + t] \\ &+ (1-u)\log[(1-t) + t\exp(\lambda)] - \lambda D\}. \end{aligned} \tag{5}$$

In Appendix E, we prove that there always exists a unique solution to the saddle-point problem defining F, so that this function is well-defined. In practice, this saddle-point problem can be solved by a variety of standard optimization methods [4]; in the work reported here, we have exploited the fact that for any fixed u, the minimization in λ can be carried out exactly in closed form (see Appendix E for details).

With these definitions, we have the following result on the effective-rate distortion function of the check-regular ensemble of LDGM codes.



(a) (b)

Fig. 2. (a) Upper bounds on the effective rate-distortion function of LDGM codes with fixed check degree γ_c. Each curve shows the rate-distortion trade-off for a different choice of the degree γ_c. All pairs (R, D) above the plotted curve are guaranteed to be achievable. (b) Dependence of the upper bound on the check degree γ_c for distortion $D \approx 0.11$, corresponding to a (Shannon) rate $R = 0.5$. Note how the required rate rapidly approaches the Shannon limit as the degree increases.

Theorem 1: Define the function $\delta^*(v;\gamma_c) := \frac{1}{2}[1-(1-2\,v)^{\gamma_c}]$. Then for any pair (R, D) satisfying the inequality

$$R > \max_{w\in[0,1]} \left[\frac{1-H(D)+F(\delta^*(w;\gamma_c),D)}{1-H(w)}\right] \tag{6}$$

there exists an integer N such that for all blocklengths $n \geq N$, we have $\frac{1}{n}\mathbb{E}_{\mathbb{C}}\mathbb{E}_S[d_n(S,\mathbb{C})] \leq D$.

Although the theorem statement is in terms of a doubly-averaged distortion (over both the source sequence S and the random code ensemble), we can also conclude that there exists a *fixed code* $\bar{\mathbb{C}}$ of suitably large blocklength n from the LDGM (γ_c) ensemble with distortion $\frac{1}{n}\mathbb{E}[d_n(S,\bar{\mathbb{C}})] \leq D$. Fig. 2(a) provides an illustration: each curve in the plot corresponds to a different choice of degree γ_c, and traces the boundary of the region of achievable (R, D) pairs guaranteed by Theorem 1. All (R, D) pairs above this curve are guaranteed to be achievable using a fixed LDGM code with the specified check degree. Of course, no code can achieve (R, D) pairs below the Shannon curve $R(D) = 1-H(D)$ also plotted in the figure. Note how the effective rate-distortion functions of the LDGM (γ_c) ensemble approach the Shannon bound as the check degree γ_c increases. Fig. 2(b) provides a more explicit illustration of the convergence to the Shannon bound: for a target distortion $D \approx 0.11$, it illustrates how the required rate approaches the Shannon bound $R(0.11) \approx 0.5$ as a function of the LDGM check degree. Note that any difference is no longer visible on the plot for check degree $\gamma_c \geq 10$.

A special case of Theorem 1 is for $D = 0$, in which case the rate-distortion problem reduces to the XORSAT problem studied in theoretical computer science, namely the problem of determining whether a given set of n parity checks (i.e., XOR constraints) over a set of m variables has a solution, known as being satisfiable. For the check-regular constructions that we have defined, the XORSAT threshold $\alpha^*(\gamma_c)$ corresponds to the maximum fraction n/m for which a randomly constructed XORSAT problem is satisfiable with high probability.

In this case, we recover as a corollary a known lower bound on the XORSAT threshold, previously established using a different analysis by Creignou *et al.* [10].

Corollary 1: The XORSAT threshold $\alpha^*(\gamma_c)$ is lower bounded as

$$\alpha^*(\gamma_c) \geq 1 \Big/ \left\{\max_{w\in[0,\frac{1}{2}]} \frac{1+\log[1-\delta^*(w;\gamma_c)]}{1-H(w)}\right\} \tag{7}$$

where $\delta^*(w;\gamma_c)$ was defined in Theorem 1.

In subsequent work, several researchers [35], [8], [12] determined the XORSAT threshold exactly, thereby showing that the bound (7), though not tight for small degrees (e.g., $\gamma_c = 3$), becomes extremely accurate for larger degrees. The bound (6) from Theorem 1, applicable to the full range of distortions $D \in [0, \frac{1}{2}]$, also exhibits a similar phenomenon. Indeed, as illustrated in Fig. 2(b), Theorem 1 provides a rigorous guarantee that the compression rate required to achieve $D^* \approx 0.11$ rapidly approaches the Shannon lower bound of $R = 0.50$ as the check degrees are increased. This rapid convergence is consistent with non-rigorous analysis based on the replica method [6].

B. Proof of Theorem 1

Having stated and discussed Theorem 1, we begin with a high-level overview of its proof. The rate-distortion statement in this theorem concerns the expected value of the random variable

$$d_n(S,\mathbb{C}) := \frac{1}{n}\min_{x\in\mathbb{C}} \|x-S\|_1, \tag{8}$$

corresponding to the (renormalized) minimum Hamming distance from a random source sequence $S \in \{0,1\}^n$ to the nearest codeword in the code $\mathbb{C}$. In informal terms, Theorem 1 states that for pairs (R, D) satisfying inequality (6), we have $\mathbb{E}_{S,\mathbb{C}}[d_n(S,\mathbb{C})] \leq D$ for sufficiently large n.

The following lemma, proved in Appendix A, guarantees that the random variable $d_n(S,\mathbb{C})$ is sharply concentrated around its expected value.

Lemma 1: For all $\epsilon > 0$,

$$\mathbb{P}[|d_n(S,\mathbb{C}) - \mathbb{E}[d_n(S,\mathbb{C})]| \geq \epsilon] \leq 2\exp(-n\epsilon^2). \qquad (9)$$

In this statement, the expectation and probability are taken both over the random source sequence and the random code. As a consequence of Lemma 1, for any target distortion D such that $D = \mathbb{E}[d_n(S,\mathbb{C})] + \epsilon$ for some $\epsilon > 0$, the probability $\mathbb{P}[d_n(S,\mathbb{C}) \leq D]$ must decay exponentially quickly in the blocklength n. Therefore, in order to prove that $\mathbb{E}[d_n(S,\mathbb{C})] \leq D$, it suffices to show that

$$\liminf_{n \to +\infty} \frac{1}{n} \log \mathbb{P}[d_n(S,\mathbb{C}) \leq D] \geq 0. \qquad (10)$$

In order to establish a lower bound of this form (10), it is equivalent to analyze the number of codewords that achieve a target distortion $D \in [0, \frac{1}{2})$. For a given LDGM code with $N = 2^{n\mathrm{R}}$ codewords, let $i = 0, 1, 2, \ldots, N-1$ be indices for the different codewords, with $i = 0$ reserved for the all-zeroes codeword. We then set the indicator random variable $Z^i(D) = 1$ if if the Hamming distance $\|X^i \oplus S\|_1$ is at most Dn, and define the random variable

$$T_n = T_n(S, \mathbb{C}; D) := \sum_{i=0}^{N-1} Z^i(D) \qquad (11)$$

that counts the number of codewords that are distortion D-good for a source sequence S. Note that $\mathbb{P}[T_n(S,\mathbb{C};D) > 0] = \mathbb{P}[d_n(S,\mathbb{C}) \leq D]$ so that it suffices to establish a bound on $\mathbb{P}[T_n > 0]$ of the form (10).

1) Bounds via Second Moment Method: Our tool for establishing the lower bound is the second moment method, a standard tool in probabilistic combinatorics [2], [1], [12], [35] that asserts[1] for any non-negative random variable T_n,

$$\mathbb{P}[T_n > 0] \geq \frac{(\mathbb{E}[T_n])^2}{\mathbb{E}\left[{T_n}^2\right]}. \qquad (12)$$

In order to apply this bound, we need to compute (or bound) the first and second moments of T_n. Beginning with the first moment, we claim that

$$\mathbb{E}[T_n] \asymp 2^{n[\mathrm{R} - (1 - H(D))]}, \qquad (13)$$

where $\asymp$ denotes equality up to subexponential factors. Indeed, by linearity of expectation, we have

$$\mathbb{E}[T_n] = \sum_{i=0}^{2^{n\mathrm{R}}-1} \mathbb{P}[Z^i(D) = 1] = 2^{n\mathrm{R}} \mathbb{P}[Z^0(D) = 1]$$

where we have used symmetry of the code construction to assert that $\mathbb{P}[Z^i(D) = 1] = \mathbb{P}[Z^0(D) = 1]$ for all indices i. Now the event $\{Z^0(D) = 1\}$ is equivalent to an i.i.d Bernoulli($\frac{1}{2}$) sequence of length n having Hamming weight less than or equal to Dn. Using standard asymptotics of binomial coefficients (see Appendix C), we have $\frac{1}{n} \log \mathbb{P}[Z^0(D) = 1] = 1 - H(D) \pm o(1)$, as claimed.

The second step is to upper bound the second moment $\mathbb{E}[{T_n}^2(D)]$, for which purpose the following lemma, proved in Appendix D, is useful. Recall that we have reserved the index $j = 0$ for the all-zeros codeword.

Lemma 2: The second moment $\mathbb{E}[{T_n}^2(D)]$ is upper bounded as

$$\mathbb{E}[T_n(D)] \left(1 + \left\{\sum_{j \neq 0} \mathbb{P}[Z^j(D) = 1 \mid Z^0(D) = 1]\right\}\right). \qquad (14)$$

Based on this lemma, we see that the key quantity to control is $\mathbb{P}[Z^j(D) = 1 \mid Z^0(D) = 1]$—corresponding to the probability that codeword X^j is distortion D-optimal $(Z^j(D) = 1)$ *given* that the all zeros-word is distortion D-optimal $(Z^0(D) = 1)$. It is this *overlap probability* that differentiates the low-density codes of interest here from the unstructured codebooks used in classical random coding arguments. In the latter case, codewords are chosen independently from some ensemble, so that the overlap probability is simply equal to $\mathbb{P}[Z^j(D) = 1]$. Thus, for the simple case of unstructured random coding, the second moment bound actually provides the converse to Shannon's rate-distortion theorem for the symmetric Bernoulli source. In contrast, for a low-density graphical code, the dependence between the events $\{Z^j(D) = 1\}$ and $\{Z^0(D) = 1\}$ requires some analysis.

2) Analysis of Overlap Probability: We now define for each $w \in [0, 1]$ the probability

$$\mathbb{Q}(w; D) := \mathbb{P}[\|X(w) \oplus S\|_1 \leq Dn \mid \|S\|_1 \leq Dn] \qquad (15)$$

where the quantity $X(w) \in \{0,1\}^n$ denotes a randomly chosen codeword, conditioned on its underlying length-m information sequence having Hamming weight $\lceil wm \rceil$. As shown in Appendix B (see Lemma 4), the random codeword $X(w)$ has i.i.d. Bernoulli elements with parameter

$$\delta^*(w; \gamma_c) = \frac{1}{2}[1 - (1 - 2\,w)^{\gamma_c}]. \qquad (16)$$

With these definitions, we now break the sum on the RHS of (14) into m terms, indexed by $t = 1, 2, \ldots, m$, where term t represents the contribution of a given non-zero information sequence $y \in \{0,1\}^m$ with (Hamming) weight t. Doing so yields

$$\sum_{j \neq 0} \mathbb{P}[Z^j(D) = 1 \mid Z^0(D) = 1] = \sum_{t=1}^{m} \binom{m}{t} \mathbb{Q}(t/m; D)$$

where we have used the fact that there are $\binom{m}{t}$ information sequences with t ones. Using standard entropy bounds on this binomial coefficient (see Appendix E), we have

$$\sum_{j \neq 0} \mathbb{P}[Z^j(D) = 1 \mid Z^0(D) = 1]$$
$$\leq m \max_{w \in [0,1]} \exp\left\{m\left[H(w) + \frac{1}{m}\log \mathbb{Q}(w; D)\right]\right\}.$$

Consequently, we need upper bounds on the probability $\mathbb{Q}(w; D)$ over the range of possible fractional weights

[1] The second-moment bound follows by applying the Cauchy-Schwarz inequality [20] to T_n and the indicator function $\mathbb{I}[T_n > 0]$ as follows:

$$(\mathbb{E}[T_n])^2 = (\mathbb{E}[T_n \mathbb{I}[T_n > 0]])^2 \leq \mathbb{E}[{T_n}^2]\mathbb{E}[\mathbb{I}^2[T_n > 0]]$$
$$= \mathbb{E}\left[{T_n}^2\right] \mathbb{P}[T_n > 0].$$

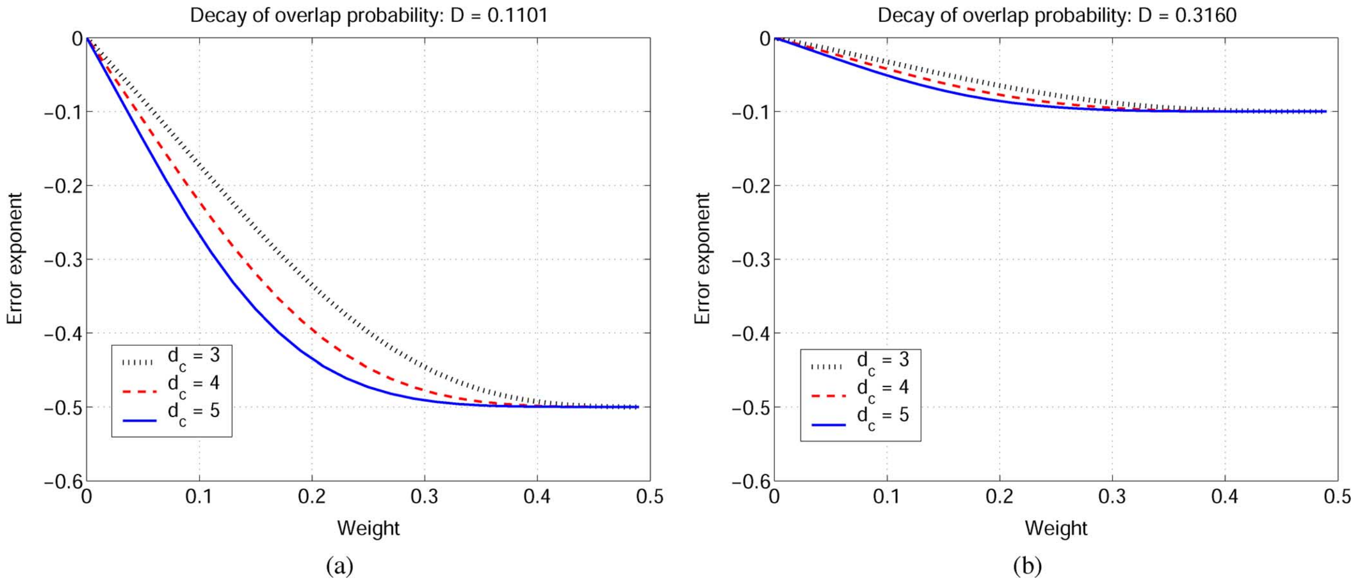


Fig. 3. Plot of the upper bound (17) on the overlap probability $\frac{1}{n}\log \mathbb{Q}(w;D)$ for different choices of the degree γ_c, and distortion probabilities. (a) Distortion $D = 0.1100$ corresponding to Shannon rate $R(D) \approx 0.5$. (b) Distortion $D = 0.3160$ corresponding to Shannon rate $R(D) \approx 0.10$.

$w \in [0,1]$. The following lemma, proved in Appendix E, provides an asymptotically exact bound on this probability. It is based on applying the Chernoff bound, and then optimizing the exponent so as to obtain the large deviations rate function.

Lemma 3: For each $w \in [0,1]$, we have

$$\frac{1}{n}\log \mathbb{Q}(w;D) \le F(\delta^*(w;\gamma_c);D) + o(1) \qquad (17)$$

where for each $t \in (0, \frac{1}{2}]$ and $D \in [0, \frac{1}{2})$, the function F was defined previously (5).

In general, for any fixed $D \in [0, \frac{1}{2})$, the function $F(t\,;D)$ has the following properties:

a) at $t = 0$, it achieves its maximum value $F(0\,;D) = 0$;
b) otherwise, it is strictly decreasing on the interval $(0, \frac{1}{2})$, approaching its minimum value $-[1 - H(D)]$ as $t \to \frac{1}{2}$.

Fig. 3 illustrates the form of the function $F(\delta^*(\omega;\gamma_c);D)$ for two different values of distortion D, and for degrees $\gamma_c \in \{3,4,5\}$. Note that increasing the check degree γ_c causes $F(\delta^*(\omega;\gamma_c);D)$ to approach its minimum $-[1 - H(D)]$ more rapidly.

3) Computing Achievable (R, D) *Pairs:* We now have the necessary ingredients to complete the proof of Theorem 1. Substituting the alternative form of $\mathbb{E}[{T_n}^2]$ from (14) into the second moment lower bound (12) yields $T_1 := \frac{1}{n}\log \mathbb{P}[T_n(D) > 0]$ is lower bounded as

$$T_1 \ge \frac{1}{n}\log \mathbb{E}[T_n(D)] \\ \underbrace{-\frac{1}{n}\log\left\{1 + \sum_{j\neq 0}\mathbb{P}[Z^j(D) = 1 \mid Z^0(D) = 1]\right\}}_{T_2}$$

On one hand, we have

$$\frac{1}{n}\log \mathbb{E}[T_n(D)] = R - (1 - H(D)) - o(1).$$

On the other hand, we have

$$T_2 \ge \frac{1}{n}\max_{w\in[0,1]}\{mH(w) + \log \mathbb{Q}(w;D)\} - o(1) \\ \ge -\max_{w\in[0,1]}\{\mathrm{R}H(w) + F(\delta^*(w;\gamma_c),D)\} - o(1)$$

where the last step follows by applying the upper bound on $\mathbb{Q}$ from Lemma 3, and the relation $\frac{m}{n} = \mathrm{R}$. Recalling that $\liminf_{n\to+\infty}\frac{1}{n}\log \mathbb{P}[T_n(D) > 0]$ is equal to $\liminf_{n\to+\infty}\frac{1}{n}\log \mathbb{P}[d_n(S,\mathbb{C}) \le D]$, we conclude

$$\lim\inf_{n\to+\infty}\frac{1}{n}\log \mathbb{P}[d_n(S,\mathbb{C}) \le D] \ge 0 \qquad (18)$$

for all rate-distortion pairs (R, D) satisfying the inequality

$$R - (1 - H(D)) \\ - \max_{w\in[0,1]}\{\mathrm{R}H(w) + F(\delta^*(w;\gamma_c),D)\} \ge 0. \qquad (19)$$

A little bit of algebra shows that condition (19) is equivalent to condition (6) in the statement of Theorem 1. Finally, by the argument following Lemma 1, condition (18) implies the claim of Theorem 1.

IV. Message-Passing Algorithms for Lossy Source Encoding

We now turn to the development of practical message-passing algorithms for performing lossy compression using low-density generator matrix codes. Like their LDPC counterparts, an attractive feature of LDGM codes is their "locally treelike" nature [41], meaning that with high probability, the neighborhood around any node will not contain any cycles of length smaller $\mathcal{O}(\log n)$. Consequently, codes are naturally suited to iterative message-passing algorithms. Given a fixed LDGM code and a randomly chosen source sequence S, the most naive approach would be to perform encoding by applying the sum-product algorithm to a given LDGM code. Unfortunately, this approach typically fails: even if the updates converge, the approximate marginals computed by sum-product typically provide little information about the optimal encoding, and the compression performance is extremely poor. Moreover, in contrast to a channel coding problem, a lossy source coding problem is typically characterized by multiple optimal (or near-optimal) encodings. Accordingly, new approaches are required for solving the lossy compression problem.

In this section, we describe message-passing algorithms for lossy compression using LDGM codes. Our approach is inspired by the success of the survey propagation algorithm [36], [34] for solving random k-SAT problems. Indeed, performing source encoding using an LDGM code is equivalent to solving a related type of satisfiability problem, known as the MAX-XORSAT problem. Like survey propagation, our approach involves two phases: 1) message-passing steps in which approximate marginal distributions are computed, and 2) decimation steps in which subsets of variables are set to their preferred values. The factor graph is then reduced by removing the set bits, and the sequence of message-passing and decimation is repeated. As in the work of Maneva *et al.* [27], we actually describe a family of algorithms: it includes decimation based on belief propagation at one extreme (BP-based decimation), and decimation based on a variant of survey propagation at the other extreme (SP-based decimation). This full range of algorithms can be obtained by varying the parameters in a single family of Markov random fields used to encode a set of generalized codewords.

More specifically, in the following sections, we begin by defining a family of Markov random fields over generalized codewords. The purpose of introducing such generalized codewords is to distinguish between "forced" variables—variables with a strong preference for some $\{0,1\}$ value in the bulk of near-optimal encodings—and free variables—those which don't exhibit any strong preferences. We then show that by suitably augmenting the state space, our Markov random fields can be captured by a factorization on the *same* factor graph as the original graph. Finally, we show that despite the state augmentation, the ordinary sum-product or belief propagation algorithm, as applied to our MRFs, can be efficiently implemented by passing only five numbers along each edge.

A. Markov Random Field Over Generalized Codewords

We begin by defining a family of Markov random fields over generalized codewords of any LDGM code, whose purpose is to represent partially assigned clusters of $\{0,1\}$-variables. Generalized codewords are members of the space $\{0,1,*\}^{n+m}$, where the interpretation of $z_i = *$ is that the associated bit i is *free*. Conversely, any bit for which $z_i \in \{0,1\}$ is *forced*. One possible view of a generalized codeword, as with the survey propagation and k-SAT problems, is as an index for a cluster of ordinary codewords.

1) Free and Forced Variables: For each check a, define its information bit neighborhood

$$N(a) := \{i \in \{1,\ldots,m\} \mid (i,a) \in E\}. \qquad (20)$$

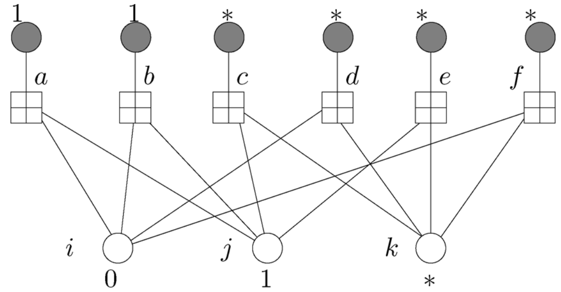


Fig. 4. Illustration of a generalized codeword for a simple LDGM. Information bits i and j are both forced; for each, the forcing checks are a and b. The remaining checks and bits are all free.

Similarly, for any bit index i, we use $N(i)$ to denote the set of its check neighbors. We then define two mutually exclusive states for a check:

i) we say that check a is forcing whenever none of its bit neighbors are free, and the local $\{0,1\}$-codeword $(x_a; z_{N(a)}) \in \{0,1\}^{1+|N(a)|}$ satisfies parity check a;

ii) on the other hand, check a is free whenever $x_a = *$, and moreover $z_i = *$ for at least one $i \in N(a)$.

We shall interchangeably say that source bit x_a is free (or forced) whenever the associated check a is free (or forcing). With this set-up, our space of generalized codewords is defined as follows.

Definition 1 (Generalized Codeword): A vector $(x,z) \in \{0,1,*\}^{n+m}$ is a valid generalized codeword when the following conditions hold:

i) all checks a are either forcing or free, and

ii) if some information bit x_i is forced (i.e., $x_i \in \{0,1\}$), then at *at least* two check neighbors $a \in N(i)$ must be forcing it.

We use $\mathbb{GC}(G)$ to denote the set of all generalized codewords defined by the generator matrix G.

For a generator matrix in which every information bit has degree two or greater, it follows from the definition that any ordinary codeword—or more precisely, any binary sequence $(x,z) \in \{0,1\}^{n+m}$ such that $x = Gz$—is also a generalized codeword. In addition, there are generalized codewords that include $*$'s in some positions, and hence do not correspond to ordinary codewords. One such (nontrivial) generalized codeword is illustrated in Fig. 4.

2) Family of Probability Distributions: We now define a family of probability distributions over generalized codewords. For any generalized codeword $(x,z) \in \{0,1,*\}^{n+m}$, define

$$N_*^{\text{sou}}(x) := |\{i \in \{1,\ldots,n\} \mid x_i = *\}| \qquad (21)$$

$$N_*^{\text{info}}(z) := |\{i \in \{1,\ldots,m\} \mid z_i = *\}| \qquad (22)$$

corresponding to the number of $*$-variables in the source and information bits, respectively. We associate nonnegative weights w_{sou} and w_{info} with the $*$-variables in the source and information bits respectively. Finally, let $\beta \geq 0$ be a parameter for penalizing disagreements between the source bits x and the given (fixed) source sequence s. With these definitions, we consider the family of probability distributions over generalized codewords of the form

$$\mathbb{P}(x, z; w_{\text{sou}}, w_{\text{info}}, \beta) \propto w_{\text{info}}^{N_*^{\text{info}}(z)} \times w_{\text{sou}}^{N_*^{\text{sou}}(x)} \times \exp^{-2\beta d_H(s,x)} \quad (23)$$

where $d_H(s, x) = \sum_{i=1}^{n} [s_i \oplus x_i]$ is the Hamming distance between the source sequence s and the reconstructed sequence x.

The family is parameterized by the triplet $(w_{\text{sou}}, w_{\text{info}}, \beta)$ of nonnegative reals, and variations of these parameters yield a range of possible distributions. For instance, in the special case $w_{\text{sou}} = w_{\text{info}} = 0$, then any generalized codeword with a free variable $*$ receives weight 0, so that the distribution (23) assigns positive mass only to ordinary codewords. For β large, any discrepancies between the reconstructed sequence x and the observed source sequence s incur strong penalty; conversely, as $\beta \to 0$ these discrepancies are penalized less heavily.

3) Markov Representation: We now describe how the family of probability distributions (23) can be represented as a Markov random field (MRF) defined on the same factor graph. An important initial observation is that the validity of a generalized codeword $x \in \{0, 1, *\}^n$ for a particular check c *cannot be determined* by looking *only* at variables in the extended check neighborhood

$$N^+(c) := \{x_c\} \cup \{x_i, i \in N(c)\}. \quad (24)$$

Rather, the validity of a generalized codeword with respect to check c depends also on all bit neighbors of checks that are incident to bits in $N(c)$, or more formally on bits with indices in the set

$$\bigcup_{i \in N(a)} \{j \in \{1, \ldots m\} | j \in N(b) \text{ for some } b \in N(i)\}. \quad (25)$$

As a particular illustration, consider the trivial LDGM code consisting of a single information bit connected to four checks (and corresponding source bits). From Definition 1, it can be seen that other than ordinary $\{0, 1\}$ codewords, the only generalized codeword is the all-$*$ configuration. Thus, any compatibility function at a for membership in the set of generalized codewords would assign zero mass to any other $\{0, 1, *\}$ configuration. Now suppose that this simple LDGM were embedded within a larger LDGM code; an instance of this embedding is provided by the information bit labeled i and corresponding checks $\{a, b, d, f\}$ in Fig. 4. With respect to the generalized codeword in this figure, we see that the local configuration $(z_i, x_a, x_b, x_d, x_f) = (0, 1, 1, *, *)$ is valid, in sharp contrast to our considerations of the trivial LDGM code in isolation. Hence, the constraints enforced by a given check change depending on the larger context in which it is embedded.

Consequently, we need to keep track of the implications of variables in the extended set (25). Accordingly, as in the work of Maneva *et al.* [27] for SAT problems, for each information bit index $i \in \{1, \ldots, m\}$, we introduce a new variable P_i, so that the 2-vector (x_i, P_i) is associated with each bit. To define P_i, first let $\mathcal{P}(i) = \mathcal{P}(N(i))$ denote the power set of all of the clause neighbors $N(i)$ of bit i. (I.e., $\mathcal{P}(i)$ is the set of all subsets of $N(i)$, and has $2^{|N(i)|}$ elements in total.) The variable P_i takes on subsets of $N(i)$, and we decompose it as $P_i = P_i^0 \cup P_i^1$, where at any time *at most one* of P_i^1 and P_i^0 are non-empty. The variable P_i has the following decomposition andinterpretation:

a) if $P_i^0 = P_i^1 = \emptyset$, then no checks are forcing bit x_i;
b) if $P_i = P_i^1 \neq \emptyset$, then certain checks are forcing x_i to be one (so that necessarily $x_i = 1$);
c) similarly, if $P_i = P_i^0 \neq \emptyset$, then certain checks are forcing x_i to be zero (so that necessarily $x_i = 0$).

By construction, this definition excludes the case that bit i is forced both by some subset of checks to be zero, and some other subset to be one (i.e., we cannot have both P_i^0 and P_i^1 non-empty at the same time). Thus, the total number of permissible states associated with the random variable P_i is $2^{|N(i)|} + 2^{|N(i)|} - 1 = 2^{|N(i)|+1} - 1$, where the subtraction of one is to avoid double-counting the empty set (case (a)).

4) Compatibility Functions: We now specify a set of compatibility functions to capture the Markov random field over generalized codewords.

a) Variable Compatibilities: For each information bit index i, the associated compatibility function takes the form: $\psi_i(z_i, P_i; w_{\text{info}})$

$$\begin{cases} 1, & \text{if } z_i = 1 \text{ and } |P_i| = |P_i^1| \geq 2 \\ 1, & \text{if } z_i = 0 \text{ and } |P_i| = |P_i^0| \geq 2 \\ w_{\text{info}}, & \text{if } z_i = * \text{ and } P_i = \emptyset. \end{cases} \quad (26)$$

On the other hand, for each source bit index a, the associated compatibility function $\psi_a(x_a; w_{\text{sou}}, \beta, s_a)$ takes the form:

$$\begin{cases} \exp(\beta), & \text{if } x_a = s_a \\ \exp(-\beta), & \text{if } x_a \neq s_a \\ w_{\text{sou}}, & \text{if } x_a = *. \end{cases} \quad (27)$$

b) Check Compatibilities: For a given check a, the associated compatibility function ψ_a^{check} is constructed to ensure that the local configuration $(x_{N(a)}, z_a)$ satisfies the local constraints to be a generalized codeword. In particular, we require the following.

a) The configuration $\{x_a\} \cup z_{N(a)}$ is *valid* for check a, meaning that (i) either it includes no $*$'s, in which case the pure $\{0, 1\}$ configuration must be a local codeword; or (ii) the associated source bit is free (i.e., $x_a = *$), and $z_i = *$ for at least one $i \in N(a)$.
b) For each index $i \in N(a)$, the following condition holds: 1) either $a \in P_i$ and a forces x_i, or 2) there holds $a \notin P_i$ and a does not force x_i.

With the singleton and factor compatibilities as above, we now define a probability distribution over the random vector

$$\{(z_i, P_i, i \in \{1, \ldots, m\}\} \bigcup \{(x_a, a \in \{1, \ldots, n\}\}.$$

For any subset A, we introduce the notation $P_A := \{P_i \mid i \in A\}$. We then define a Markov random field (MRF) on the original factor graph in the following way:

$$\begin{aligned} &\widetilde{\mathbb{P}}((z, P, x; w_{\text{sou}}, w_{\text{info}}, \beta, s) \\ &\propto \prod_{i=1}^{m} \psi_i(z_i, P_i; w_{\text{info}}) \\ &\quad \times \prod_{a=1}^{n} \psi_a(x_a; \beta, w_{\text{sou}}, s_a) \\ &\quad \times \prod_{a=1}^{n} \psi_a^{\text{check}}(z_{N(a)}, P_{N(a)}, x_a). \end{aligned} \quad (28)$$

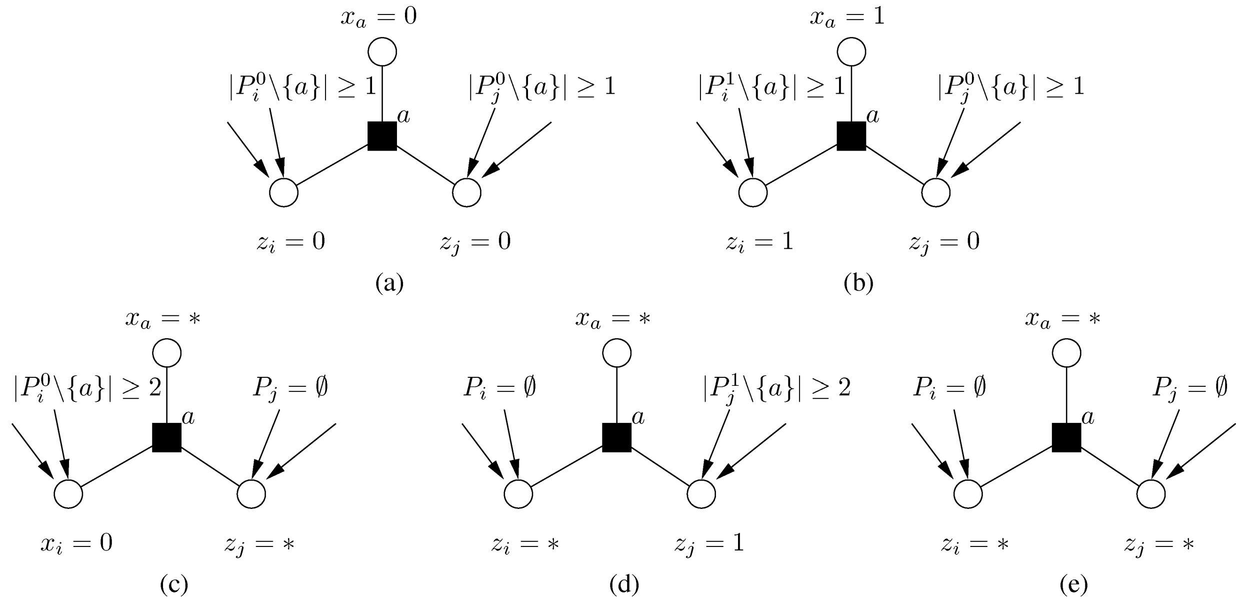


Fig. 5. Illustration of different settings for $(z_{N(a)}, x_a)$ for it to be a valid part of a generalized codeword. (a) Check a is forcing $(x_a = 0)$, so that each of z_i and z_j must have at least one other forcing check in their neighborhood $(|P_i^0\backslash\{a\}| \geq 1)$. (b) Analogous set-up with check a forcing with $x_a = 1$. (c) Check a is free $(x_a = *)$. At least one of z_i or z_j must be free. Since $z_i = 0$, it must be forced by at least two checks other than a (i.e., $|P_i^0\backslash\{a\}| \geq 2$). (d) Similar set-up to (c), in which $z_i = *$ is now free, and $z_j = 1$ must be forced by at least two checks other than a (i.e., $|P^1{}_j\backslash\{a\}| \geq 2$). (e) Check a is free, and both bits z_i and z_j are also free. We must have $P_i = P_j = \emptyset$.

Recall that $N(a)$ denotes the set of neighbors of check a in the factor graph. Moreover, the quantities (z, P, x) are random variables, whereas the triplet $(w_{\text{sou}}, w_{\text{info}}, \beta) \in \mathbb{R}^3$ and the fixed source sequence $s \in \{0,1\}^n$ parameterize the model.

Proposition 1: When marginalized down to the (z, x) variables, the marginal $\widetilde{\mathbb{P}}(z, x; w_{\text{sou}}, w_{\text{info}}, \beta, s)$ of the Markov random field (28) agrees with the weighted distribution (23).

Proof: The proof entails some book-keeping to keep track of various terms. Let us fix some configuration $(z, x) \in \{0, 1, *\}^{m+n}$. The product $\prod_{a=1}^n \psi_a^{\text{check}}(z_{N(a)}, P_{N(a)}, x_a)$ of check compatibility functions ensures that the string (z, x) is locally consistent with respect to the checks. More specifically, for each check a, there are two possibilities.

i) If $x_a \neq *$, then we must have $P_a = \{a\}$, $a \notin P_{N(a)}$, and no $*$ symbols in $z_{N(a)}$, and with $(x_a, z_{N(a)})$ a valid local codeword. In this case, the variable compatibility function ψ_a contributes a factor of $\exp(-\beta)$ if $x_a \neq s_a$, and a factor of $\exp(\beta)$ if $x_a = s_a$.

ii) If $x_a = *$, then the check compatibility enforces the constraint $P = \emptyset$, and the variable compatibility ψ_a contributes a factor of w_{sou}.

Summarizing the contributions to the probability distribution from the source compatibilities ψ_a and check compatibilities ψ_a^{check}, we see that they contribute a total weight of

$$\prod_{a=1}^n w_{\text{sou}}^{\mathbb{I}[x=*]} \times \prod_{a=1}^n \exp(-\beta[s_a \oplus x_a])$$
$$\times \prod_{a=1}^n \exp(\beta[1 - x_a \oplus s_a])$$
$$\propto w_{\text{sou}}^{N_*^{\text{sou}}(x)} \times \exp^{-2\beta d_H(s,x)}.$$

Finally, the product of information bit compatibilities ψ_i ensures that any information bits from the configuration (z, x) with $x_i \in \{0, 1\}$ is forced by at least two checks. (If not, the configuration is assigned a weight of zero). In the case that $x_i = *$, the probability distribution picks up a weight w_{info}. Overall, we recover the weighting specified in (23).

B. Family of Belief Propagation Algorithms

One consequence of Proposition 1 is that the marginal distributions over information bits x_i of the weighted distribution (23) are equivalent to marginals of the Markov random field (28), defined on the *same* factor graph as the original LDGM code. When the LDGM degrees are bounded, such factor graphs are well-known [41] to be "locally treelike", so that it is reasonable to seek approximations to the marginal distributions $\widetilde{\mathbb{P}}_i = \{\widetilde{\mathbb{P}}_i(0), \widetilde{\mathbb{P}}_i(1), \widetilde{\mathbb{P}}_i(*)\}$ in this extended MRF using the usual sum-product message-passing updates. In this section, we describe how these message-passing updates can nonetheless be performed efficiently, despite some obstacles that are initially present.

In our extended Markov random field, the random variable at each information variable node i is of the form (z_i, P_i), and belongs to the Cartesian product $\{0, 1, *\} \times [\mathcal{P}(i) \times \{0, 1\}]$. (To clarify the additional $\{0, 1\}$, the variable $P_i = P_i^0 \cup P_i^1$ corresponds to a particular subset of $\mathcal{P}(i)$, but we also need to specify whether $P_i = P_i^0$ or $P_i = P_i^1$.) Since the cardinality of $\mathcal{P}(i)$ can be large (i.e., exponential in the number of check neighbors of a given bit), it might appear difficult to implement the message-passing in an efficient manner. Fortunately, it turns out to be necessary to keep track of only five numbers—regardless of the number of check neighbors—in order to characterize a message on any given edge (whether from check to bit or from bit to check). These five cases are illustrated in Fig. 5, and defined more formally as follows.

$$M^{0f}_{i\to a} \leftarrow \psi_i(0)\Big\{\prod_{b\in N(i)\backslash\{a\}}\big[M^{0f}_{b\to i}+M^{0w}_{b\to i}\big]-\prod_{b\in N(i)\backslash\{a\}}M^{0w}_{b\to i}\Big\} \tag{29a}$$

$$M^{1f}_{i\to a} \leftarrow \psi_i(1)\Big\{\prod_{b\in N(i)\backslash\{a\}}\big[M^{1f}_{b\to i}+M^{1w}_{b\to i}\big]-\prod_{b\in N(i)\backslash\{a\}}M^{1w}_{b\to i}\Big\} \tag{29b}$$

$$M^{0w}_{i\to a} \leftarrow \psi_i(0)\Big\{\prod_{b\in N(i)\backslash\{a\}}\big[M^{0f}_{b\to i}+M^{0w}_{b\to i}\big]-\prod_{b\in N(i)\backslash\{a\}}M^{0w}_{b\to i}-\sum_{c\in N(i)\backslash\{a\}}M^{0f}_{c\to i}\prod_{b\in N(i)\backslash\{a,c\}}M^{0w}_{b\to i}\Big\}. \tag{29c}$$

$$M^{1w}_{i\to a} \leftarrow \psi_i(1)\Big\{\prod_{b\in N(i)\backslash\{a\}}\big[M^{1f}_{b\to i}+M^{1w}_{b\to i}\big]-\prod_{b\in N(i)\backslash\{a\}}M^{1w}_{b\to i}-\sum_{c\in N(i)\backslash\{a\}}M^{1f}_{c\to i}\prod_{b\in N(i)\backslash\{a,c\}}M^{1w}_{b\to i}\Big\}. \tag{29d}$$

$$M^{*}_{i\to a} \leftarrow \omega^*_i\prod_{b\in N(i)\backslash\{a\}}M^{*}_{b\to i} \tag{29e}$$

Fig. 6. Messages from bits to checks.

$$M^{0f}_{a\to i} \leftarrow \frac{1}{2}\Big[\prod_{j\in N(a)\backslash\{i\}}\big(M^{0f}_{j\to a}+M^{1f}_{j\to a}\big)+\prod_{j\in N(a)\backslash\{i\}}\big(M^{0f}_{j\to a}-M^{1f}_{j\to a}\big)\Big] \tag{30a}$$

$$M^{1f}_{a\to i} \leftarrow \frac{1}{2}\Big[\prod_{j\in N(a)\backslash\{i\}}\big(M^{0f}_{j\to a}+M^{1f}_{j\to a}\big)-\prod_{j\in N(a)\backslash\{i\}}\big(M^{0f}_{j\to a}-M^{1f}_{j\to a}\big)\Big] \tag{30b}$$

$$M^{0w}_{a\to i} \leftarrow \prod_{j\in N(a)\backslash\{i\}}\big[M^{*}_{j\to a}+M^{1w}_{j\to a}+M^{0w}_{j\to a}\big]-\prod_{j\in N(a)\backslash\{i\}}\big[M^{1w}_{j\to a}+M^{0w}_{j\to a}\big] \tag{30c}$$

$$-\sum_{k\in N(a)\backslash\{i\}}M^{*}_{k\to a}\prod_{j\in N(a)\backslash\{i,k\}}\big[M^{1w}_{j\to a}+M^{0w}_{j\to a}\big] \tag{30d}$$

$$M^{1w}_{a\to i} = M^{0w}_{a\to i}. \tag{30e}$$

$$M^{*}_{a\to i} \leftarrow \prod_{j\in N(a)\backslash\{i\}}\big[M^{*}_{j\to a}+M^{1w}_{j\to a}+M^{0w}_{j\to a}\big]-\prod_{j\in N(a)\backslash\{i\}}\big[M^{1w}_{j\to a}+M^{0w}_{j\to a}\big] \tag{30f}$$

Fig. 7. Messages from checks to bits.

i) $(z_i = 0, a \in P_i^0)$: In this case, check a is forcing z_i to be equal to zero, so that we say that z_i is a *forced zero with respect to* a. We use the notation $M^{0f}_{i\to a}$ and $M^{0f}_{a\to i}$ for the corresponding bit-to-check and check-to-bit messages. In Fig. 5(a), both z_i and z_j are forced zeros with respect to a.

ii) $(z_i = 1, a \in P_i^1)$: Similarly in this case, check a is forcing z_i to be equal to one, so that we say that z_i is a *forced one with respect to* a, and denote the corresponding messages with $M^{1f}_{i\to a}$ and $M^{1f}_{a\to i}$. For instance, in Fig. 5(b), z_i is a forced one with respect to a.

iii) $(z_i = 0, \emptyset \neq P_i^0 \subseteq F(i)\backslash\{a\})$: Some subset of checks *not* including a is forcing $z_i = 0$, in which case we say that z_i is a *weak zero with respect to check a*. In this case, the messages are denoted by $M^{0w}_{i\to a}$ and $M^{0w}_{a\to i}$. In Fig. 5(c), z_i is a weak zero with respect to a.

iv) $(z{=}1, \emptyset \neq P_i^1 \subseteq F(i)\backslash\{a\})$: Similarly, some subset of checks *not* including a forces $z_i = 1$, in which case we say that z_i is a *weak one with respect to check a*, and use the corresponding messages $M^{1w}_{i\to a}$ and $M^{1w}_{a\to i}$. In Fig. 5(d), z_i is a weak one with respect to a.

v) $(z_i = *, P_i^1 = P_i^0 = \emptyset)$: No checks force bit z_i; associated messages are denoted by $M^{*}_{i\to a}$ and $M^{*}_{i\to a}$. In Fig. 5(e), both z_i and z_j are free, and not forced by any check.

With these definitions, it is straightforward (but requiring some calculation) to derive the BP message-passing updates as applied to the generalized MRF, as shown in Figs. 6 and 7. In these equations, we think of applying the bit-to-check updates to both information bits (for $j \in \{1,\ldots,m\}$), and source bits (for $a \in \{1,\ldots,n\}$). In both cases, we simply specify the bit by a generic index i. When i indexes a source bit (say a), the message updates are trivial, since the source bit simply relays the information in its compatibility function ψ_a to its associated check. The quantities $\psi_i(1)$ and $\psi_i(0)$ are shorthand for values of the full compatibility function (ψ_i for information bits, and ψ_a for source bits). Finally, the parameter $\omega^*{}_i$ is equal to w_{info} for an information bit, and w_{sou} for a source bit. See Appendix F for a detailed derivation of these message-passing updates.

1) Computation of Pseudomarginals: We are now equipped with update rules for the sum-product algorithm, as applied to the extended Markov random field (28), for computing approximations to the marginal distributions over the information bits. Assuming that the message updates converge, the sum-product pseudomarginals (i.e., approximations to the true marginal distributions) are calculated as follows:

$$\mu_i(0) \propto \psi_i(0)\Bigg\{\prod_{a\in N(i)}\Big[M^{0f}_{a\to i}+M^{0w}_{a\to i}\Big] - \prod_{a\in N(i)}M^{0w}_{a\to i}-\sum_{b\in N(i)}M^{0f}_{b\to i}\prod_{a\in N(i)\backslash\{b\}}M^{0w}_{a\to i}\Bigg\} \tag{31a}$$

$$\mu_i(1) \propto \psi_i(1) \Bigg\{ \prod_{a \in N(i)} \left[M^{1f}_{a \to i} + M^{1w}_{a \to i} \right] - \prod_{a \in N(i)} M^{1w}_{a \to i} - \sum_{b \in N(i)} M^{1f}_{b \to i} \prod_{a \in N(i) \setminus \{b\}} M^{1w}_{a \to i} \Bigg\} \tag{31b}$$

$$\mu_i(*) \propto \omega^*{}_i \prod_{a \in N(i)} M^{*}_{a \to i}. \tag{31c}$$

See Appendix G for a derivation of the form of these pseudomarginals. The overall triplet is normalized to sum to one.

2) Decimation Procedure: As with the survey propagation algorithm applied to satisfiability problems [36], [27], we use the pseudomarginals computed by sum-product as an engine to drive a sequence of decimation steps. More specifically, we use a form of hard decimation in which a fraction of information bits are set to their preferred binary values after each round of message-passing, according to the following procedure:

1) Run the message-passing algorithm until convergence (up to a pre-specified tolerance), and compute the pseudomarginals according to (31).
2) Set a subset of information bits based on their biases. $B_i := |\mu_i(1) - \mu_i(0)|$. In particular, we rank order the biases by their absolute value, and set a fraction α of the information bits to their preferred value

$$z_i = \begin{cases} 1, & \text{if } B_i > 0 \\ 0, & \text{if } B_i \leq 0. \end{cases}$$

3) Simplify the resulting code as follows:
   a) remove all hard-set information bits z_i from the graph, and remove all checks not connected to information bits;
   b) while there exists any check b with less than two information bit neighbors, do the following recursively: for check b connected to information bit z_j, set $z_j = s_b$, and remove check b and bit j from the graph.
4) Return to step 1) until all information bits have been set.

C. Experimental Results

In this section, we report the results of various experiments that we have conducted. We have applied a C-based implementation of our message-passing and decimation algorithm to LDGM codes with blocklengths ranging from $n = 256$ to $n = 100000$ source bits. In all experiments reported here, we generated LDGM codes by forming the duals of "good" LDPC codes. (This choice, though heuristic, is not unreasonable given the rigorous connection between channel decoding over the binary erasure channel using LDPC codes, and source encoding in the binary erasure quantization problem using LDGM codes [32].) We found empirical performance to be best using LDGM codes with irregular degree distributions, such as the degree distributions that emerge from optimization based on density evolution [41], optimized for the binary symmetric channel. In contrast, for regular LDGM codes (e.g., a rate $R = 0.5$ code with constant check degree 3 and bit degree 6), the message-passing algorithm fails to build up appreciable biases until a large number of bits have been set to their preferred values. As noted by a reviewer, a specific important question is to gain a deeper understanding of the degree distributions to which the message-passing and decimation approach are well-suited. In our experiments, we found empirically that a certain fraction of degree two checks were necessary for the message-passing steps to build up appreciable biases. (If decimation decisions are made on the basis of weak biases, then they are likely to be unreliable and lead to poor source encoding.) Intuitively, as with belief propagation over the erasure channel, having a reasonable fraction of degree two checks ensures that decimation successively produces checks with degree one, which will be highly biased. We note that the LDGM codes that we used, as the duals of good LDPC codes, do indeed have a significant fraction of degree two checks. Currently, we do not have a good understanding of the minimum fraction required for the problem of lossy source coding, in contrast to the case of channel coding.

We have described a family of message-passing/decimation algorithms, parameterized by the pair $(w_{\text{sou}}, w_{\text{info}})$, which range from BP-based decimation ($w_{\text{sou}} = w_{\text{info}} = 0$), to various forms of message-passing on generalized codewords ($w_{\text{sou}}, w_{\text{info}} > 0$). Here we report some representative results with $w_{\text{sou}} = \exp(0.10)$ and $w_{\text{info}} = \exp(0.05)$, which in our experiments yielded stable message-passing and good source encoding performance. We also parameterized the decimation procedure using two additional parameters: in each decimation round, we set all bits with biases above a threshold t, up to a maximum percentage of $p \in [0, 1]$ bits. In practice, we observed that our algorithm required a constant number of decimation rounds to terminate; since each round of message-passing requires $\mathcal{O}(n)$ operations, the algorithm runs in $\mathcal{O}(n)$ time. In the implementation given here, the algorithm requires about two minutes for a problem with blocklength $n = 10\,000$; however, our implementation is inefficient, and could be substantially accelerated. In practice, for any fixed blocklength n, the parameters t and p can have a substantial effect on running time, and we explore the resulting trade-offs with distortion performance below. We refer the interested reader to follow-up work by Filler and Fridrich [14], [15], which includes a thorough set of experimental results for a wide range of parameter settings and decimation schemes.

In Fig. 8(a), we plot the empirical rate-distortion performance of codes with blocklengths $n = 10\,000$, over a range of rates. As the rate varied, we varied the parameter β—controlling the degree of fidelity to the source sequence—from $\beta = 1.45$ for rate $R = 0.90$ down to $\beta = 0.65$ for rate $R = 0.30$. Intuitively, one expects such a monotonic relationship between β and R, since codes of higher rate should be able to match a higher number of the source bits (lower distortion).[2] In each round of each trial, we set all information bits with biases above the threshold $t = 0.70$, up to a maximum percentage $p = 0.05$ of the total number of bits. Each diamond plotted in the figure represents (for a fixed rate) the average distortion over 15 trials. Note that the performance is already very close to the Shannon

[2] Note, however, that we do not have any formal justification for setting β. We note that work by Ciliberti *et al.* [6], [7] has suggested that β should be chosen based a Legendre transformation linking "complexity" to rate; we refer the reader to these papers for more details.

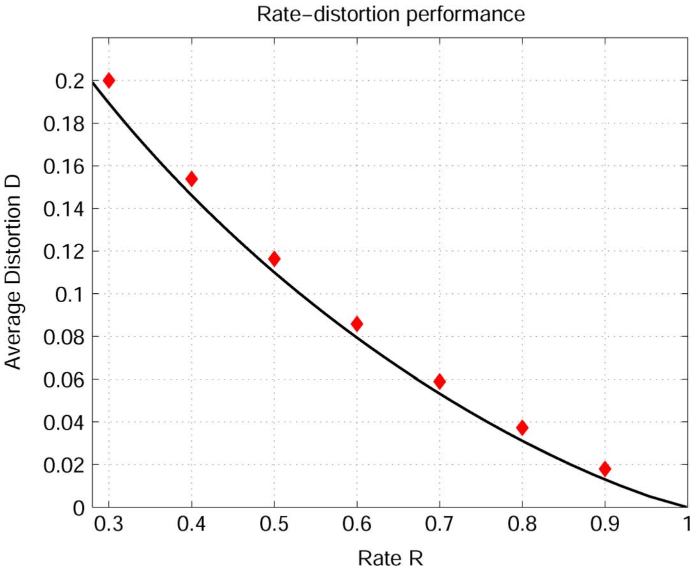


Fig. 8. Plot of rate versus distortion, comparing the Shannon limit (solid line) and empirical performance using LDGM codes with blocklength $n = 10\,000$. Each diamond is the average distortion over 15 trials.

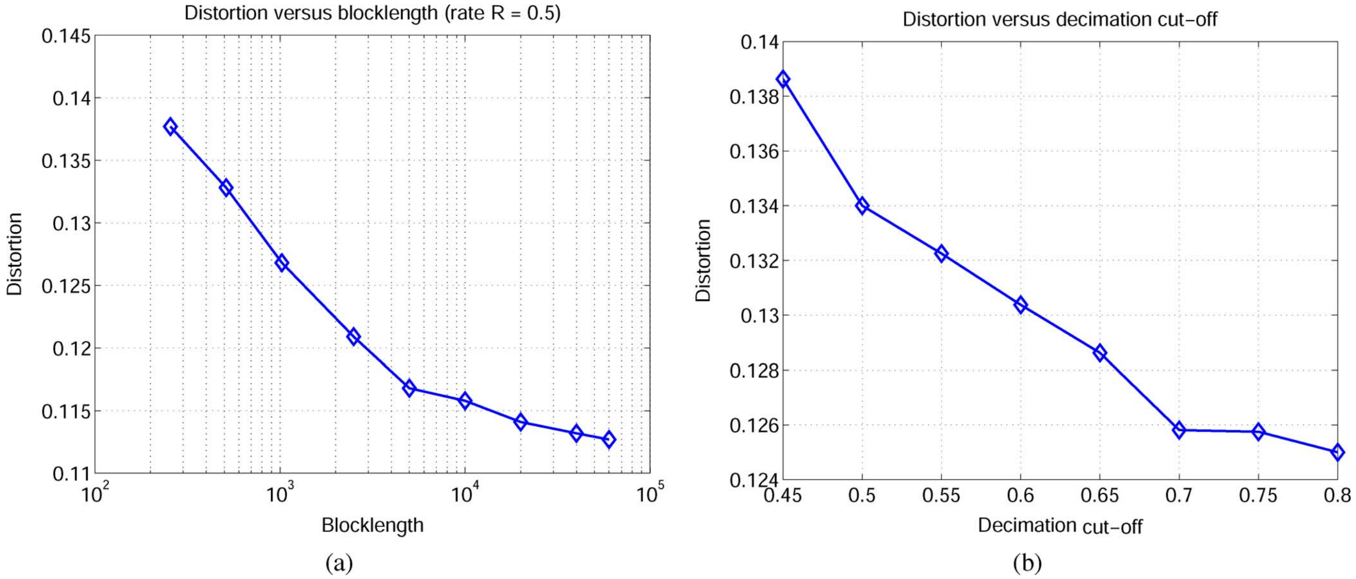


Fig. 9. (a) Plot of distortion versus blocklength n, for fixed rate $R = 0.5$, with blocklengths ranging from $n = 256$ to $n = 60\,000$. Performance is mediocre for blocklengths less than $n = 1000$. At blocklength $n = 10\,000$, the achieved distortion D is roughly 0.115 and decreases more slowly towards the Shannon limit as blocklength is further increased. (b) Plots of distortion versus the aggressiveness of the decimation procedure, as controlled by the cut-off parameter t on the required bias for a bit to be set.

rate-distortion bound (lower curve), even for intermediate block length $n = 10\ 000$.

In the next set of experiments, we explored how (for a fixed rate) the distortion performance varies as we changed code parameters such as the blocklength n, and algorithmic parameters. In Fig. 9(a), we fix the rate $R = 0.50$, and plot the distortion over blocklengths ranging from $n = 256$ to $n = 60\ 000$, using the same algorithmic parameters as before. The distortion decreases to its minimum of $d \approx 0.1128$ for $n = 60\ 000$, in comparison to the ultimate Shannon limit $d^* \approx 0.11$. As noted above, the aggressiveness of the decimation procedure can be controlled by the threshold t on the minimum bias required before setting bits. Fig. 9(b) shows plots of the distortion versus the threshold t, with lower values of t corresponding to a more aggressive decimation procedure. These plots correspond to a fixed code of rate $R = 0.50$ and blocklength $n = 1600$; in each round of decimation, we set all bits with biases above t, up to a maximum percentage of $p = 0.20$ of bits. The obtained distortion decreases monotonically as t is increased corresponding to a more conservative procedure; beyond the threshold $t \approx 0.70$, there appear to be limited gains to being even more conservative in decimation. There is about a five-fold increase in computation time in moving from $t = 0.45$ to $t = 0.70$.

Last, we compare the results obtained by message-passing and decimation with LDGM codes to the results of trellis-coded quantization [28], where the lossy source encoding problem can

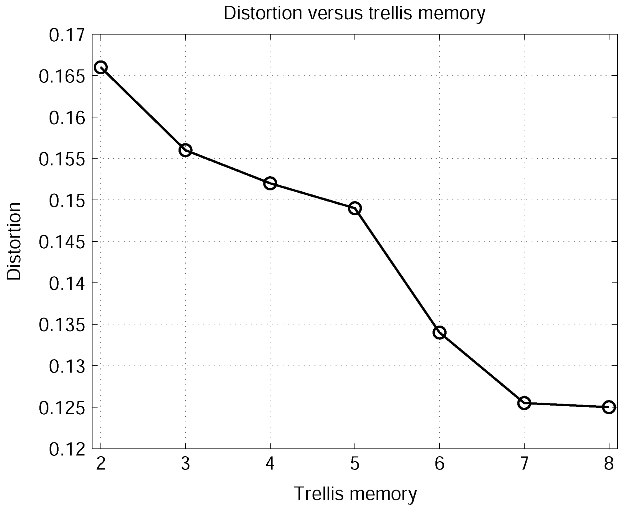


Fig. 10. Plots of distortion D versus the constraint length (memory) in trellis-coded quantization. Trellis codes were taken from the paper [28]; for each given constraint length, we performed 10 trials with a block length $n = 10\ 000$.

be solved using the Viterbi algorithm on the associated trellis code. In this case, improved distortion performance is obtained by increasing the constraint length or memory of the trellis code, as opposed to the blocklength for LDGM codes; note that the computational complexity of the Viterbi algorithm scales exponentially in the trellis memory. Fig. 10 shows results for the trellis-coded quantization (TCQ) method, using rate $R = 0.50$ trellis codes taken from the paper [28]; the plots show the obtained distortion, averaged over 10 trials with blocklength $n = 100\ 000$, for a sequence of constraint lengths ($\ell = 2, 3, \ldots, 8$). The obtained distortion decreases monotonically until it appears to saturate at constraint length $\ell = 7$, where $D \approx 0.125$. This performance is similar to that obtained in Fig. 9 for LDGM codes with blocklengths $n = 1600$, and the empirical running time of the TCQ method (in our current implementations) is faster. However, we were unable to find trellis codes that achieve distortion substantially below $D \approx 0.12$ for rate $R = 0.50$ codes, as do our LDGM codes with blocklength $n = 10\ 000$ in Fig. 8. One might be able to do so with more refined searches for trellis parameters, or by increasing the constraint length; however, we note that increasing the constraint length incurs an exponential penalty in running time, so that constraint lengths much beyond $\ell = 8$ are not feasible.

V. Conclusion and Future Directions

In this paper, we have studied the use of low-density generator matrix (LDGM) codes for lossy source encoding. On the theoretical side, we established rigorous upper bounds on the distortion performance of these codes under optimal minimum distance encoding. This analysis establishes that the rate-distortion tradeoff achievable by these codes rapidly approaches the Shannon limit as the check degrees are increased. On the practical side, we proposed a family of message-passing algorithms, ranging from ordinary belief propagation over codewords to a variant of survey propagation. Such message-passing algorithms can be combined with decimation steps in order to perform effective yet practical source encoding, as we illustrated via empirical simulations.

Our work suggests a number of open questions for future research. First, although the current paper has only analyzed the performance of the check-regular ensemble of LDGM codes, it should be possible to extend our moment analysis technique to more general ensembles of codes. Second, it remains to gain theoretical understanding of the message-passing/decimation algorithms proposed in this paper and other work [36], [27], [6], [7]. Analysis of message-passing combined with decimation appears challenging, since the act of decimation introduces statistical dependencies that invalidate the standard assumptions underlying standard methods like density evolution. Accordingly, an important research direction is to explore alternative methods for analyzing algorithms that involve some form of decimation. Third, although the current paper focused only on the Bernoulli symmetric source, it should be possible to apply similar constructions and techniques for lossy source encoding of other discrete and continuous sources.

Appendix I

A. *Proof of Lemma 1*

Our proof exploits the standard vertex-exposure martingale [37] in order to establish that the random variable $d_n(S, \mathbb{C})$ is sharply concentrated around its expectation. Consider a fixed sequential labelling $\{1, \ldots, n\}$ of LDGM checks, with checks i associated with source bit S_i. We reveal the check and associated source bit in a sequential manner for each $i = 1, \ldots, n$, and so define a sequence of random variables $\{U_0, U_1, \ldots, U_n\}$ via $U_0 := \mathbb{E}[d_n(S, \mathbb{C})]$, and

$$U_i := \mathbb{E}[d_n(S, \mathbb{C}) \mid S_1, \ldots, S_i], \qquad i = 1, \ldots, n. \tag{32}$$

Note that we have $U_n = d_n(S, \mathbb{C})$ by construction. Moreover, this sequence satisfies the following bounded difference property: adding any source bit S_i and the associated check in moving from U_{i-1} to U_i can lead to a (renormalized) change in the minimum distortion of at most $c_i = 1/n$. Consequently, by applying Azuma's inequality [2], we have, for any $\epsilon > 0$

$$\mathbb{P}[|d_n(S, \mathbb{C}) - \mathbb{E}[d_n(S, \mathbb{C})]| \geq \epsilon] \leq \exp(-n\epsilon^2). \tag{33}$$

B. *Basic Property of LDGM Codes*

For a given weight $w \in (0, 1)$, suppose that we enforce that the information sequence $y \in \{0, 1\}^m$ has exactly $\lceil wm \rceil$ ones. Conditioned on this event, we can then consider the set of all codewords $X(w) \in \{0, 1\}^n$, where we randomize over low-density generator matrices G chosen as in step (a) above. Note for any fixed code, $X(w)$ is simply some codeword, but becomes a random variable when we imagine choosing the generator matrix G randomly. The following lemma characterizes this distribution as a function of the weight w and the LDGM top degree γ_c:

Lemma 4: Given a binary vector $y \in \{0, 1\}^m$ with a fraction w of ones, the distribution of the random LDGM codeword $X(w)$ induced by y is an i.i.d. Bernoulli sequence with parameter $\delta^*(w; \gamma_c) = \frac{1}{2}\left[1 - (1 - 2\,w)^{\gamma_c}\right]$.

Proof: Given a fixed sequence $y \in \{0,1\}^m$ with a fraction w ones, the random codeword bit $X_i(w)$ at bit i is formed by connecting γ_c edges to the set of information bits.[3] Each edge acts as an i.i.d. Bernoulli variable with parameter w, so that we can write

$$X_i(w) = V_1 \oplus V_2 \oplus \ldots \oplus V_{\gamma_c} \tag{34}$$

where each $V_k \sim \operatorname{Ber}(w)$ is independent and identically distributed. A straightforward calculation using z-transforms (see [17]) or Fourier transforms over $\mathrm{GF}(2)$ yields that $X_i(w)$ is Bernoulli with parameter $\delta^*(w;\gamma_c)$ as defined.

C. Bounds on Binomial Coefficients

The following bounds on binomial coefficients are standard (see [9, Ch. 12]); for any positive integer n and all integers k such that $1 \leq k \leq n$, we have

$$H\left(\frac{k}{n}\right) - \frac{\log(n+1)}{n} \leq \frac{1}{n}\log\binom{n}{k} \leq H\left(\frac{k}{n}\right) \tag{35}$$

where $H(\cdot)$ is the binary entropy function (see (2)).

D. Proof of Lemma 2

First, by the definition of $T_n(D)$, we have

$$\begin{aligned}\mathbb{E}[{T_n}^2(D)] &= \mathbb{E}[\sum_{i=1}^{N-1}\sum_{j=0}^{N-1} Z^i(D)Z^j(D)] \\ &= \mathbb{E}[T_n] + \sum_{i=0}^{N-1}\sum_{j\neq i}\mathbb{P}[Z^i(D)=1,\, Z^i(D)=1].\end{aligned}$$

To simplify the second term on the RHS, we first note that for any i.i.d Bernoulli($\frac{1}{2}$) sequence $S \in \{0,1\}^n$ and any codeword X^j, the binary sequence $S' := S \oplus X^j$ is also i.i.d. Bernoulli($\frac{1}{2}$). Consequently, for each pair $i \neq j$, the probability $p = \mathbb{P}[Z^i(D)=1,\, Z^j(D)=1]$ takes the form

$$\begin{aligned}p &= \mathbb{P}[\|X^i \oplus S\|_1 \leq Dn, \|X^j \oplus S\|_1 \leq Dn] \\ &= \mathbb{P}[\|X^i \oplus S'\|_1 \leq Dn, \|X^j \oplus S'\|_1 \leq Dn] \\ &= \mathbb{P}[\|X^i \oplus X^j \oplus S\|_1 \leq Dn, \|S\|_1 \leq Dn].\end{aligned}$$

Note that for each $j \neq i$, the vector $X^i \oplus X^j$ is a non-zero codeword. For each fixed i, summing over $j \neq i$ can be recast as summing over all non-zero codewords, so that the sum $s = \sum_{i\neq j}\mathbb{P}[Z^i(D)=1,\, Z^j(D)=1]$ takes the form

$$\begin{aligned}s &= \sum_{i=0}^{N-1}\sum_{j\neq i}\mathbb{P}[\|X^i \oplus X^j \oplus S\|_1 \leq Dn, \|S\|_1 \leq Dn] \\ &= \sum_{i=0}^{N-1}\sum_{k\neq 0}\mathbb{P}[\|X^k \oplus S\|_1 \leq Dn, \|S\|_1 \leq Dn] \\ &= 2^{n\mathrm{R}}\sum_{k\neq 0}\mathbb{P}[\|X^k \oplus S\|_1 \leq Dn, \|S\|_1 \leq Dn] \\ &= 2^{n\mathrm{R}}\mathbb{P}[Z^0(D)=1]\sum_{k\neq 0}\mathbb{P}[Z^k(D)=1 \mid Z^0(D)=1] \\ &= \mathbb{E}[T_n]\sum_{k\neq 0}\mathbb{P}[Z^k(D)=1 \mid Z^0(D)]\end{aligned}$$

thus establishing the claim.

E. Proof of Lemma 3

We reformulate the probability $\mathbb{Q}(w,D)$ as follows. Recall that $\mathbb{Q}$ involves conditioning the source sequence S on the event $\|S\|_1 \leq Dn$. Accordingly, we define a discrete variable T with distribution

$$\mathbb{P}(T=t) = \frac{\binom{n}{t}}{\sum_{s=0}^{Dn}\binom{n}{s}} \quad \text{for } t = 0,1,\ldots,Dn$$

representing the (random) number of "1"s in the source sequence S. Let U_i and V_j denote Bernoulli random variables with parameters $1-\delta^*(w;\gamma_c)$ and $\delta^*(w;\gamma_c)$ respectively. With this set-up, conditioned on codeword j having a fraction wn ones, the quantity $\mathbb{Q}(w,D)$ is equivalent to the probability that the random variable

$$W := \begin{cases}\sum_{i=1}^{T} U_j + \sum_{j=1}^{n-T} V_j, & \text{if } T \geq 1 \\ \sum_{j=1}^{n} V_j, & \text{if } T = 0\end{cases} \tag{36}$$

is less than Dn. To bound this probability, we use a Chernoff bound in the form

$$\frac{1}{n}\log\mathbb{P}[W \leq Dn] \leq \inf_{\lambda<0}\left(\frac{1}{n}\log\mathbb{M}_W(\lambda) - \lambda D\right) \tag{37}$$

where $\mathbb{M}_W(\lambda) = \mathbb{E}[\exp(\lambda W)]$ is the moment generating function. Taking conditional expectations and using independence, we have

$$\mathbb{M}_W(\lambda) = \sum_{t=0}^{Dn}\mathbb{P}[T=t][\mathbb{M}_U(\lambda)]^t[\mathbb{M}_V(\lambda)]^{n-t}.$$

Here the cumulant generating functions have the form

$$\log\mathbb{M}_U(\lambda) = \log[(1-\delta)e^{\lambda} + \delta], \quad \text{and} \tag{38a}$$

$$\log\mathbb{M}_V(\lambda) = \log[(1-\delta) + \delta e^{\lambda}] \tag{38b}$$

where we have used (and will continue to use) δ as a shorthand for $\delta^*(w;\gamma_c)$.

Of interest to us is the exponential behavior of this expression in n. Using the standard entropy approximations to the binomial coefficient (see Appendix C), we have the bound $\mathbb{M}_W(\lambda) \leq f(n)\sum_{t=0}^{Dn} g(t)$, where $f(n)$ is a polynomial function, and

$$\begin{aligned}g(t) := \exp\Bigg[n\Bigg\{&H\left(\frac{t}{n}\right) - H(D) \\ &+ \frac{t}{n}\log\mathbb{M}_U(\lambda) + \left(1-\frac{t}{n}\right)\log\mathbb{M}_V(\lambda)\Bigg\}\Bigg].\end{aligned} \tag{39}$$

Further analyzing this sum, the quantity $\frac{1}{n}\log\sum_{t=0}^{Dn} g(t)$ is upper bounded by

$$\begin{aligned}\frac{1}{n}\max_{0\leq t\leq Dn}\log g(t) + \frac{\log f(n)}{n} + \frac{\log(nD)}{n}& \\ = \frac{1}{n}\max_{0\leq t\leq Dn}\log g(t) + o(1).&\end{aligned}$$

[3] In principle, our procedure allows two different edges to choose the same information bit, but the probability of such double-edges is asymptotically negligible.

The quantity $T_1 := \frac{1}{n}\max_{0\le t\le Dn}\log g(t)$ is upper bounded by

$$\max_{0\le t\le Dn}\left\{H\left(\frac{t}{n}\right) - H(D) + \frac{t}{n}\log \mathbb{M}_U(\lambda) + \left(1-\frac{t}{n}\right)\log \mathbb{M}_V(\lambda)\right\}$$

and hence by the problem

$$\max_{u\in[0,D]}\{H(u)-H(D)+u\log\mathbb{M}_U(\lambda)+(1-u)\log\mathbb{M}_V(\lambda)\}.$$

Combining this upper bound on $\frac{1}{n}\log\mathbb{M}_W(\lambda)$ with the Chernoff bound (37) yields that

$$\frac{1}{n}\log\mathbb{P}[W\le Dn] \le \inf_{\lambda<0}\max_{u\in[0,D]} G(u,\lambda;\delta)+o(1) \qquad (40)$$

where the function G takes the form

$$H(u)-H(D)+u\log\mathbb{M}_U(\lambda)+(1-u)\log\mathbb{M}_V(\lambda)-\lambda D.$$

Finally, we establish that the solution (u^*,λ^*) to the min-max saddle point problem (40) is unique, and specified by $u^*=D$ and λ^* as in Lemma 3. First of all, observe that for any $\delta\in(0,1)$, the function G is continuous, strictly concave in u and strictly convex in λ. (The strict concavity follows since $H(u)$ is strictly concave with the remaining terms linear; the strict convexity follows since cumulant generating functions are strictly convex.) Therefore, for any fixed $\lambda<0$, the maximum over $u\in[0,D]$ is always achieved. On the other hand, for any $D>0$, $u\in[0,D]$ and $\delta\in(0,\frac{1}{2})$, we have $G(u;\lambda;t)\to+\infty$ as $\lambda\to-\infty$, so that the infimum is either achieved at some $\lambda^*<0$, or at $\lambda^*=0$. Thus far, using standard saddle point theory [22], we have established the existence and uniqueness of the saddle point solution (u^*,λ^*).

To verify the fixed point conditions, we compute partial derivatives in order to find the saddle point. First, considering u, we compute

$$\begin{aligned}\frac{\partial G}{\partial u}(u,\lambda;\delta) &= \log\frac{1-u}{u}+\log\mathbb{M}_U(\lambda)-\log\mathbb{M}_V(\lambda)\\ &= \log\frac{1-u}{u}+\log[(1-\delta)e^{\lambda}+\delta]\\ &\quad-\log[(1-\delta)+\delta e^{\lambda}].\end{aligned}$$

Solving the $\frac{\partial G}{\partial u}(u,\lambda;\delta)=0$ yields the condition

$$u'=\frac{\exp(\lambda)}{1+\exp(\lambda)}(1-\delta)+\frac{1}{1+\exp(\lambda)}\delta \qquad (41)$$

which is always non-negative. If in addition $u'<D$, then it is optimal; otherwise, the optimum is achieved at $u=D$.

Turning now to the minimization over λ, we compute the partial derivative to find

$$\begin{aligned}\frac{\partial G}{\partial\lambda}(u,\lambda;\delta) &= u\frac{(1-\delta)\exp(\lambda)}{(1-\delta)\exp(\lambda)+\delta}\\ &\quad+(1-u)\frac{\delta\exp(\lambda)}{(1-\delta)+\delta\exp(\lambda)}-D.\end{aligned}$$

Setting this partial derivative to zero yields a quadratic equation in $\exp(\lambda)$ with coefficients

$$a=\delta\,(1-\delta)\,(1-D) \qquad (42a)$$

$$b=u(1-\delta)^2+(1-u)\delta^2-D[\delta^2+(1-\delta)^2]. \qquad (42b)$$

$$c=-D\delta(1-\delta). \qquad (42c)$$

Since we need $\exp(\lambda)>0$ for all λ, we must take the unique positive root of this quadratic equation.

F. Detailed Derivation of Message Updates

1) Updating Messages From Bit to Check: We first the detailed form of the message updates from bit to check.

a) Force-Zero Message: Suppose that bit i want to send a a force zero message. Since we necessarily have $z_i=0$ if a is forcing i, there should be a pre-factor $\psi_i(0)$ corresponding to the weight assigned to $z_i=0$. Second, the messages from all other checks $b\in N(i), b\neq a$ should be either force zero or weak zero messages, with at least one force zero message coming from a check in $N(i)\backslash\{a\}$, and we should sum over all possible partitions into the two message types. Using the binomial formula, this reasoning leads to

$$M^{0f}_{i\to a}=\psi_i(0)\left\{\prod_{b\in N(i)\backslash\{a\}}\left[M^{0f}_{b\to i}+M^{0w}_{b\to i}\right]-\prod_{b\in N(i)\backslash\{a\}}M^{0w}_{b\to i}\right\}. \qquad (43)$$

b) Force-One Message: Completely analogous reasoning yields

$$M^{1f}_{i\to a}=\psi_i(1)\left\{\prod_{b\in N(i)\backslash\{a\}}\left[M^{1f}_{b\to i}+M^{1w}_{b\to i}\right]-\prod_{b\in N(i)\backslash\{a\}}M^{1w}_{b\to i}\right\}. \qquad (44)$$

c) Weak-Zero Message: Now consider the case of i sending a a weak zero message. In this case, the only way for $z_i=0$ is for z_i to be forced by 2 checks $b\in N(i)\backslash\{a\}$. (Recall that by definition i can send a weak message only if a is not forcing i.) Thus, we need to exclude the case of all one messages from neighbors $b\in N(i)\backslash\{a\}$. These considerations lead to the following sum:

$$M^{0w}_{i\to a}=\psi_i(0)\sum_{S\subseteq N(i)\backslash\{a\},|S|\ge 2}\prod_{b\in S}M^{0f}_{b\to i}\prod_{b\in S^c}M^{0w}_{b\to i}$$

Using the binomial formula, we can write

$$M^{0w}_{i\to a}=\psi_i(0)\left\{\prod_{b\in N(i)\backslash\{a\}}\left[M^{0f}_{b\to i}+M^{0w}_{b\to i}\right]-\prod_{b\in N(i)\backslash\{a\}}M^{0w}_{b\to i}-\sum_{c\in N(i)\backslash\{a\}}M^{0f}_{c\to i}\prod_{b\in N(i)\backslash\{a,c\}}M^{0w}_{b\to i}\right\}. \qquad (45)$$

d) Weak-One Message: The equation for a weak one is entirely analogous:

$$M^{1w}_{i\to a}=\psi_i(0)\left\{\prod_{b\in N(i)\backslash\{a\}}\left[M^{0f}_{b\to i}+M^{0w}_{b\to i}\right]\right.$$

$$- \prod_{b\in N(i)\backslash\{a\}} M^{0w}_{b\to i} - \sum_{c\in N(i)\backslash\{a\}} M^{0f}_{c\to i} \prod_{b\in N(i)\backslash\{a,c\}} M^{0w}_{b\to i} \Bigg\}. \tag{46}$$

e) All $*$ Message: Finally, for bit i to send a $*$-message, it must receive all $*$ messages from its other bit neighbors:

$$M^{*}_{i\to a} = \omega^{*}{}_{i} \prod_{b\in N(i)\backslash\{a\}} M^{*}_{b\to i} \tag{47}$$

2) Updating Messages From Checks to Bits: For each check a, let $\mathbf{C}(a)$ be the associated set of local codewords

$$\mathbf{C}(a) := \{z_{N(a)} \in \{0,1\}^{|N(a)|} | \text{ check } a \text{ satisfied}\}. \tag{48}$$

a) Strong-0/1 Messages: Check a will send bit i a force zero message only for configurations that are valid local codewords with $z_i = 0$, which yields

$$M^{0f}_{a\to i} = \sum_{\{z\in\mathbf{C}(a) | z_i = 0\}} \prod_{j\in N(a)\backslash\{i\}} \left[M^{0f}_{j\to a}\right]^{(1-z_j)} \left[M^{1f}_{j\to a}\right]^{(z_j)} \tag{49}$$

Similar reasoning yields the update for the force one message: $M^{1f}_{a\to i}$ is updated to

$$\sum_{\{z\in\mathbf{C}(a) | z_i = 1\}} \prod_{j\in N(a)\backslash\{i\}} \left[M^{0f}_{j\to a}\right]^{(1-z_j)} \left[M^{1f}_{j\to a}\right]^{(z_j)}. \tag{50}$$

As in the ordinary sum-product algorithm for LDPC codes, these summations can be simplified considerably via the use of the fast Fourier transform over GF(2). Given a function $f : \{0,1\} \to \mathbb{C}$, recall that its Fourier transform $\hat{f} = \mathcal{F}(f)$ is a function $\hat{f} : \{0,1\} \to \mathbb{C}$ is given by

$$\mathcal{F}(f)(y) = \begin{cases} f(0) + f(1) & \text{if } y = 0 \\ f(0) - f(1) & \text{if } y = 1 \end{cases}$$

Moreover, we have the inverse relationship $F^{-1}(\hat{f}) = f$, where

$$\mathcal{F}^{-1}(\hat{f})(x) = \begin{cases} \frac{1}{2}[\hat{f}(0) + \hat{f}(1)], & \text{if } x = 0 \\ \frac{1}{2}[\hat{f}(0) - \hat{f}(1)], & \text{if } x = 1 \end{cases}$$

Let us index the set $N(a)\backslash\{i\}$ by elements $\{1,\ldots,d\}$ and define $f_j(x_j) = [M^{0f}_{j\to a}]^{(1-x_j)}[M^{1f}_{j\to a}]^{(x_j)}$ for each $j = 1,\ldots,d$. With this notation, we can write the summation defining $M^{0f}_{a\to i}$ as a convolution

$$\begin{aligned} M^{0f}_{a\to i} &= (f_1 * f_2 * \cdots * f_d)(0) \\ &= \mathcal{F}^{-1}\left(\prod_{j=1}^{d} \hat{f}_j\right)(0) \\ &= \frac{1}{2}\left[\prod_{j=1}^{d} \hat{f}_j(0) + \prod_{j=1}^{d} \hat{f}_j(1)\right] \end{aligned}$$

where we have used the Fourier relation between convolution and products. For each $j = 1,\ldots,d$, we have

$$\hat{f}(y) = \begin{cases} M^{0f}_{j\to a} + M^{1f}_{j\to a}, & \text{if } y = 0 \\ M^{0f}_{j\to a} - M^{1f}_{j\to a}, & \text{if } y = 1 \end{cases}$$

Thus, from (51), the strong-0 message $M^{0f}_{a\to i}$ is updated to

$$\frac{1}{2}\left[\prod_{j=1}^{d}\left(M^{0f}_{j\to a} + M^{1f}_{j\to a}\right) + \prod_{j=1}^{d}\left(M^{0f}_{j\to a} - M^{1f}_{j\to a}\right)\right] \tag{51}$$

A similar equation holds for the strong-1 update

$$M^{1f}_{a\to i} = \frac{1}{2}\left[\prod_{j=1}^{d}\left(M^{0f}_{j\to a} + M^{1f}_{j\to a}\right) - \prod_{j=1}^{d}\left(M^{0f}_{j\to a} - M^{1f}_{j\to a}\right)\right]. \tag{52}$$

b) Weak-0/1 Messages: To send a weak zero message, the check a must have at least two stars among $N(a)\backslash\{i\}$, and the remaining variables can send either weak one or weak zero messages, which yields

$$M^{0w}_{a\to i} = \sum_{S\subseteq N(a)\backslash\{i\}, |S|\geq 2} \prod_{j\in S} M^{*}_{j\to a} \prod_{j\in S^c} \left[M^{1w}_{j\to a} + M^{0w}_{j\to a}\right].$$

Using the binomial formula, we find that $M^{0w}_{a\to i}$ is equal to

$$\begin{aligned} &\prod_{j\in N(a)\backslash\{i\}} \left[M^{*}_{j\to a} + M^{1w}_{j\to a} + M^{0w}_{j\to a}\right] \\ &\quad - \prod_{j\in N(a)\backslash\{i\}} \left[M^{1w}_{j\to a} + M^{0w}_{j\to a}\right] \\ &\quad - \sum_{k\in N(a)\backslash\{i\}} M^{*}_{k\to a} \prod_{j\in N(a)\backslash\{i,k\}} \left[M^{1w}_{j\to a} + M^{0w}_{j\to a}\right]. \end{aligned}$$

The weak one message update is identical $M^{1w}_{a\to i} = M^{0w}_{a\to i}$.

c) All $*$ Message: Finally, for check a to send a $*$-message, it must receive at least one other $*$, and the other messages can be weak ones or zeros, so that $M^{*}_{a\to i}$ is equal to

$$\sum_{S\subseteq N(a)\backslash\{i\}, |S|\geq 1} \prod_{j\in S} M^{*}_{j\to a} \prod_{j\in S^c} \left[M^{1w}_{j\to a} + M^{0w}_{j\to a}\right]. \tag{53}$$

Again using the binomial formula, we rewrite this update as

$$\begin{aligned} M^{*}_{a\to i} = &\prod_{j\in N(a)\backslash\{i\}} \left[M^{*}_{j\to a} + M^{1w}_{j\to a} + M^{0w}_{j\to a}\right] \\ &- \prod_{j\in N(a)\backslash\{i\}} \left[M^{1w}_{j\to a} + M^{0w}_{j\to a}\right]. \end{aligned} \tag{54}$$

G. Formulae for Pseudomarginals

Finally, we derive the equations for pseudomarginals in our Markov random field. First, a variable z_i can be equal to zero if and only if at least two neighbors are sending it strong zero messages, and the remaining neighbors are sending weak zero messages. Thus, we have

$$\mu_i(0) \propto \psi_i(0) \sum_{S\subseteq N(i), |S|\geq 2} \prod_{a\in S} M^{0f}_{a\to i} \prod_{a\in S^c} M^{0w}_{a\to i}$$

$$= \psi_i(0) \left\{ \prod_{a \in N(i)} \left[M^{0f}_{a \to i} + M^{0w}_{a \to i} \right] - \prod_{a \in N(i)} M^{0w}_{a \to i} - \sum_{b \in N(i)} M^{0f}_{b \to i} \prod_{a \in N(i) \setminus \{b\}} M^{0w}_{a \to i} \right\} \tag{55}$$

where the second equality follows from the binomial formula.

The expression for $\mu_i(1)$ is entirely analogous

$$\mu_i(1) \propto \psi_i(1) \sum_{S \subseteq N(i), |S| \geq 2} \prod_{a \in S} M^{1f}_{a \to i} \prod_{a \in S^c} M^{1w}_{a \to i}$$

$$= \psi_i(1) \left\{ \prod_{a \in N(i)} \left[M^{1f}_{a \to i} + M^{1w}_{a \to i} \right] - \prod_{a \in N(i)} M^{1w}_{a \to i} - \sum_{b \in N(i)} M^{1f}_{b \to i} \prod_{a \in N(i) \setminus \{b\}} M^{1w}_{a \to i} \right\}. \tag{56}$$

Finally, the only term contributing to $\mu_i(*)$ is when all incoming messages are $*$.

$$\mu_i(*) \propto \omega^*{}_i \prod_{a \in N(i)} M^*_{a \to i}. \tag{57}$$

Acknowledgment

The authors would like to thank the anonymous reviewers for their careful reading and constructive suggestions.

References

[1] D. Achlioptas and Y. Peres, "The threshold for random k-SAT is $2k \log 2 - o(k)$," *J. AMS*, vol. 17, pp. 947–973, 2004.

[2] N. Alon and J. Spencer, *The Probabilistic Method*. New York: Wiley Interscience, 2000.

[3] C. Berroux and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[4] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[5] S.-Y. Chung, G. D. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[6] S. Ciliberti and M. Mézard, "The theoretical capacity of the parity source coder," *J. Stat. Mechan.*, p. P10003, 2006.

[7] S. Ciliberti, M. Mézard, and R. Zecchina, "Message-passing algorithms for non-linear nodes and data compression," *Complexus*, vol. 3, p. 58, 2006.

[8] S. Cocco, O. Dubois, J. Mandler, and R. Monasson, "Rigorous decimation-based construction of ground pure states for spin-glass models on random lattices," *Phys. Rev. Lett.*, vol. 90, no. 4, Jan. 2003.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[10] N. Creignou, H. Daudé, and O. Dubois, "Approximating the satisfiability threshold of random XOR formulas," *Combin., Probab. Comput.*, vol. 12, pp. 113–126, 2003.

[11] A. G. Dimakis, M. J. Wainwright, and K. Ramchandran, Lower Bounds on the Rate-Distortion Function of LDGM Codes Aug. 2008, Tech. Rep., Univ. California, Berkeley.

[12] O. Dubois and J. Mandler, "The 3-XORSAT threshold," in *Proc. 43rd Symp. FOCS*, 2002, pp. 769–778.

[13] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inf. Theory*, vol. 51, pp. 954–972, Mar. 2005.

[14] T. Filler, "Minimizing Embedding Impact in Steganography Using Low Density Codes," Master's thesis, SUNY Binghamton, , Jul. 2007.

[15] T. Filler and J. Fridrich, "Binary quantization using belief propagation with decimation over factor graphs of LDGM codes," in *Proc. Allerton Conf. Commun. Contr. Computing*, Sept. 2007.

[16] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proc. SPIE Electron. Imag.*, Jan. 2007, vol. 66, pp. 1–15.

[17] R. G. Gallager, *Low-Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[18] J. Garcia-Frias and Y. Zhao, "Compression of binary memoryless sources using punctured turbo codes," *IEEE Commun. Lett.*, vol. 6, no. 9, pp. 394–396, Sept. 2002.

[19] T. Goblick, "Coding for a Discrete Information Source With a Distortion Measure" Ph.D. dissertation, MIT, , 1962 [Online]. Available: http://hdl.handle.net/1721.1/39984

[20] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*. Oxford, U.K.: Oxford Science, Clarendon, 1992.

[21] A. Gupta and S. Verdu, "Nonlinear sparse graph codes for lossy compression," in *Proc. Int. Symp. Inf. Theory Workshop*, Lake Tahoe, NV, 2007.

[22] J. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms Volume 1*. New York: Springer-Verlag, 1993.

[23] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[24] S. Kudekar and R. Urbanke, Lower Bounds on the Rate-Distortion Function of Individual LDGM Codes Tech. Rep. arXiv:cs.IT:0804.1697, Apr. 2008, EPFL.

[25] H. A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Mag.*, vol. 21, pp. 28–41, 2004.

[26] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman, "Improved low-density parity check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, pp. 585–598, Feb. 2001.

[27] E. Maneva, E. Mossel, and M. J. Wainwright, "A new look at survey propagation and its generalizations," *J. ACM*, vol. 54, no. 4, pp. 2–41, 2007.

[28] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 82–93, 1990.

[29] E. Martinian and M. J. Wainwright, "Analysis of LDGM and compound codes for lossy compression and binning," in *Proc. Workshop Inf. Theory Applicat. (ITA)*, Feb. 2006, pp. 229–233, Available at arxiv:cs.IT/0602046.

[30] E. Martinian and M. J. Wainwright, "Low density codes achieve the rate-distortion bound," in *Proc. Data Compress. Conf.*, Mar. 2006, vol. 1, pp. 153–162, Available at arxiv:cs.IT/061123.

[31] E. Martinian and M. J. Wainwright, "Low density codes can achieve the Wyner-Ziv and Gelfand-Pinsker bounds," in *Proc. Int. Symp. Inf. Theory*, July 2006, pp. 484–488, Available at arxiv:cs.IT/0605091.

[32] E. Martinian and J. S. Yedidia, "Iterative quantization using codes on graphs," in *Proc. Allerton Conf. Contr., Comput. Commun.*, Oct. 2003.

[33] Y. Matsunaga and H. Yamamoto, "A coding theorem for lossy data compression by LDPC codes," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2225–2229, 2003.

[34] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems," *Science*, 2002.

[35] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina, "Alternative solutions to diluted p-spin models and XORSAT problems," *J. Statist. Phys.*, vol. 111, no. 105, 2002.

[36] M. Mézard and R. Zecchina, "Random k-satisfiability: From an analytic solution to an efficient algorithm," *Phys. Rev. E*, vol. 66, no. 056126, 2002.

[37] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge University Press, 1995.

[38] T. Murayama, "Thouless-Anderson-Palmer approach for lossy compression," *Phys. Rev. E*, vol. 69, pp. 035105(1)–035105(4), 2004.

[39] T. Murayama and M. Okada, "One step RSB} scheme for the rate distortion function," *J. Phys. A: Math. Gen.*, vol. 65, pp. 11123–11130, 2003.

[40] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.

[41] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity check codes," *IEEE Trans. Inf. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[42] T. Richardson and R. Urbanke, *Modern Coding Theory*. New York: Cambridge University Press, 2008.

[43] D. Schonberg, S. S. Pradhan, and K. Ramchandran, "LDPC codes can approach the Slepian-Wolf bound for general binary sources," in *Proc. 40th Annu. Allerton Conf. Contr., Commun., Comput.*, Oct. 2002, pp. 576–585.

[44] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, pp. 533–547, Sep. 1980.

[45] A. J. Viterbi and J. K. Omura, "Trellis encoding of memoryless discrete-time sources with a fidelity criterion," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 3, pp. 325–332, 1974.

[46] M. J. Wainwright and E. Maneva, "Lossy source coding by message-passing and decimation over generalized codewords of LDGM codes," in *Proc. Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 2005, Available at arxiv:cs.IT/0508068.

[47] M. J. Wainwright and E. Martinian, "Low-density codes that are optimal for binning and coding with side information," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1061–1079, Mar. 2009.

[48] A. G. Dimakis, M. J. Wainwright, and K. Ramchandran, "Lower bounds on the rate-distortion function of LDGM codes," presented at the Int. Symp. Inf. Theory Workshop (ITW) 2007, Lake Tahoe, NV, 2007.

Martin Wainwright received the Ph.D. degree in electrical engineering and computer science (EECS) from Massachusetts Institute of Technology (MIT), Cambridge.

He is currently an Associate Professor at University of California at Berkeley, with a joint appointment between the Department of Statistics and the Department of Electrical Engineering and Computer Sciences. His research interests include statistical signal processing, coding and information theory, statistical machine learning, and high-dimensional statistics.

Dr. Wainwright has been awarded an Alfred P. Sloan Foundation Fellowship, an NSF CAREER Award, the George M. Sprowls Prize for his dissertation research, a Natural Sciences and Engineering Research Council of Canada 1967 Fellowship, and IEEE Signal Processing Society Best Paper Award in 2008.

Elitza Maneva received the Bachelor's degree in engineering and applied science from the California Institute of Technology (Caltech), Pasadena, and the Ph.D. degree in computer science from the University of California, Berkeley in 2006. After graduation, she worked as a postdoctoral researcher at the IBM Almaden Research Center before joining the Polytechnical University of Catalunya (UPC). She is now with the Institut de Matematica, Universitat de Barcelona (UB), Barcelona, Spain.

Emin Martinian received the B.S. degree from the University of California, Berkeley, in 1997 and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, in 2000 and 2004, respectively, all in electrical engineering and computer science.

Since 2006, he has been with Bain Capital, Boston, MA, where he works on research and strategy development at Bain's global macro hedge fund, Absolute Return Capital. In addition, he is affiliated with the Systems, Information, and Algorithms Laboratory in the Research Laboratory of Electronics at MIT, where he is involved in problems of information theory, prediction, and machine learning. Prior to his current position, he worked at research laboratories including Mitsubishi Electric Research Laboratories, Cambridge, MA, and Lucent Bell Laboratories, Murray Hill, NJ, and at technology startups including PinPoint Corporation and OPC Technologies. He holds more than 10 patents.

Dr. Martinian received the Capocelli Best Paper Award at the Data Compression Conference and second place for best Computer Science Ph.D. dissertation at MIT.