

Transactions Letters

A Modified Belief Propagation Algorithm for Code Word Quantization

Phillip A. Regalia, *Fellow, IEEE*

Abstract—Modern coding advances, including dirty paper coding and information hiding, require quantizing a given binary word to a code word. A ‘good’ solution would approach the rate-distortion bound in lossy source compression. Here we propose a simple variant on belief propagation which is observed to converge to a solution giving respectable rate-distortion performance. Comparisons with other recently proposed source quantization methods reveal that the proposed algorithm holds particular interest in short block-length applications, as encountered in packet-based communication systems.

Index Terms—Source compression, code word quantization, rate-distortion theory, information hiding, dirty paper coding.

I. INTRODUCTION

THE general decoding problem for linear binary codes is to deduce a minimum-weight error pattern consistent with a given error syndrome. This problem arose in early studies of error correction codes since, under reasonable channel assumptions, the minimum weight correction to an error-prone received block of bits gives the maximum likelihood estimate of the true code word sent. Although conceptually straightforward, the general decoding problem actually belongs to the class of NP-complete problems [1], [2], [3], thus challenging the development of efficient algorithms for its resolution. The hard aspect of this problem also motivates its use in cryptography (see, e.g., [4]) and underlies heuristic search procedures for low-weight error patterns to assess cryptographic strength [5], [6], [2], [7].

The general decoding problem has met with resurgent interest in modern coding applications, including dirty paper coding for multi-user communications [8], information hiding with robustness [9], [10], [11], and wet paper coding in steganography [12], [13], [14]. In these more recent applications, the error syndrome is replaced by constraints deriving from side information (typically messages to be hidden or users to be accommodated), thus exposing the duality with source coding and quantization [9], [10], [11], [15]. In this vein, recent algorithmic work has focused on approaching the rate-distortion bound for lossy source compression (e.g., [16]–[22]) using variants of belief propagation combined possibly

with heuristic search procedures. Various reported results (such as [18], [19], [22]) indeed approach the theoretical rate-distortion curve, albeit at an appreciable level of algorithmic complexity.

The intent of this note is to propose a simple modification to the standard belief propagation algorithm which, when used in a low-density generator matrix configuration, is observed to yield a convergent algorithm for code word quantization, offering respectable rate-distortion performance. The resulting algorithm is significantly simpler than the survey propagation algorithm [16] and its variants [18]–[21], [23], or even pruning techniques applied to standard belief propagation [24], thus rendering it potentially suitable for real-time applications. At the same time, its performance displays consistent behavior over variations in block length and choice of generator matrix as compared to the message-passing algorithm of Murayama [25], although the latter is arguably still to be preferred in the long block-length case.

II. BACKGROUND

Let \mathbf{H} be an $(N-K) \times N$ parity check matrix comprised of ones and zeros, operating on the Galois field $\text{GF}(2)$ (modulo-2 arithmetic over integers). The set of vectors $\mathbf{c} \in [\text{GF}(2)]^N$ in its null space defines a linear code \mathcal{C} of rate K/N :

$$\mathbf{H}\mathbf{c} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{c} \in \mathcal{C}.$$

The set of vectors $\mathbf{y} \in [\text{GF}(2)]^N$ which instead produce a prescribed syndrome $\mathbf{s} \in [\text{GF}(2)]^{N-K}$ comprise the coset $\mathcal{C}(\mathbf{s})$ for that syndrome:

$$\mathcal{C}(\mathbf{s}) = \{\mathbf{y} \in [\text{GF}(2)]^N : \mathbf{H}\mathbf{y} = \mathbf{s}\}.$$

The code \mathcal{C} induced by \mathbf{H} is simply the coset for the zero syndrome: $\mathcal{C} = \mathcal{C}(\mathbf{0})$. Note that with $\mathbf{y} \in \mathcal{C}(\mathbf{s})$ and $\mathbf{c} \in \mathcal{C}$, their modulo-2 sum remains in the coset: $\mathbf{y} + \mathbf{c} \in \mathcal{C}(\mathbf{s})$.

Suppose we are given an N -element binary vector $\mathbf{w} \in [\text{GF}(2)]^N$ and consider the problem of finding an N -element binary vector $\mathbf{y} \in [\text{GF}(2)]^N$ which minimizes the Hamming distance $d(\mathbf{w}, \mathbf{y})$ subject to $(N-K)$ parity constraints:

$$\min_{\mathbf{y}} d(\mathbf{w}, \mathbf{y}), \quad \text{subject to } \mathbf{s} = \mathbf{H}\mathbf{y} \quad (1)$$

This problem arises in information hiding [9], [10], [11], dirty paper coding [8] (once transcribed to its $\text{GF}(2)$ setting [10]), and wet paper coding [12], [13].

If the bits in \mathbf{w} hail from a uniform source, then from rate-distortion theory [26], the minimum average distortion

$$D_{\min} = \frac{1}{2^N} \sum_{\mathbf{w} \in [\text{GF}(2)]^N} \min_{\mathbf{y} \in \mathcal{C}(\mathbf{s})} \frac{d(\mathbf{w}, \mathbf{y})}{N}$$

Paper approved by M. Skoglund, the Editor for Source/Channel Coding of the IEEE Communications Society. Manuscript received May 5, 2008; revised January 21, 2009 and April 24, 2009.

Parts of this work were presented at the Asilomar Conference on Circuits, Systems and Computers, Nov. 2007. This work was supported by the National Science Foundation under grant CCF 0634757.

P. A. Regalia is with the Department of Electrical Engineering and Computer Science, Catholic University of America, Washington, DC 20064 USA (e-mail: regalia@cua.edu).

Digital Object Identifier 10.1109/TCOMM.2009.12.080050

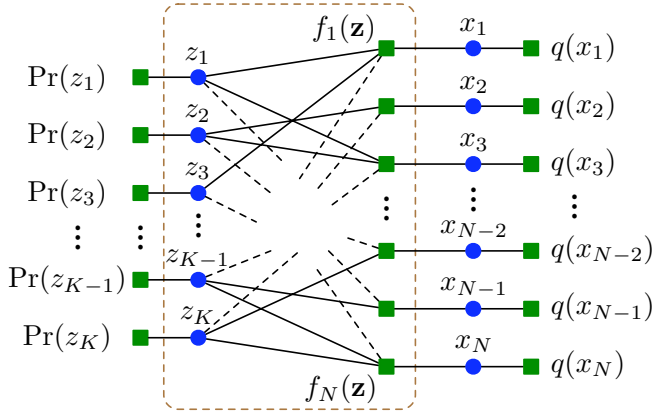


Fig. 1. Factor graph for a low density generator matrix.

relates to the code rate K/N via the rate-distortion curve [26]

$$H_2(D_{\min}) \geq 1 - (K/N) \quad (2)$$

where $H_2(p) = -p \log p - (1-p) \log(1-p)$ is the binary entropy function.

Finding a closest \mathbf{y} is equivalent to finding a minimum weight vector consistent with the syndrome:

$$\min_{\mathbf{e}} d(\mathbf{e}, \mathbf{0}) \quad \text{subject to} \quad \mathbf{s} = \mathbf{H}\mathbf{e}$$

This vector \mathbf{e} is the *coset leader* of $\mathcal{C}(\mathbf{s})$, and the solution to (1) becomes $\mathbf{y} = \mathbf{w} + \mathbf{e}$ (using the modulo-2 sum). This minimum weight problem, in turn, is known to belong to the class of NP-complete problems [1], [6], [2], [3], and thus the decoding problem (1) is hard.

Conventional belief propagation decoding applied to this problem, with \mathbf{w} a “received” vector, \mathbf{s} the “side information,” and \mathbf{y} the “hidden measurement” (as elucidated in [17]), does not in general converge to a meaningful solution, unless \mathbf{w} is fortuitously within a decoding vicinity of $\mathcal{C}(\mathbf{s})$ (e.g., [21]). As such, other approaches must be pursued.

A promising approach is offered by the low-density generator matrix formulation (e.g., [27], [28], [20], [21]). Let \mathbf{G} be an $N \times K$ generator matrix for the code $\mathcal{C}(\mathbf{0})$, so that each code word $\mathbf{c} \in \mathcal{C}(\mathbf{0})$ may be expressed as

$$\mathbf{c} = \mathbf{G}\mathbf{z}, \quad \text{for some } K\text{-bit vector } \mathbf{z}.$$

If \mathbf{x} is a particular solution satisfying the parity constraint $\mathbf{s} = \mathbf{H}\mathbf{x}$, then $\mathbf{x} + \mathbf{c}$ also satisfies the parity constraint for any $\mathbf{c} \in \mathcal{C}(\mathbf{0})$, and indeed, all elements of the coset $\mathcal{C}(\mathbf{s})$ may be so expressed. As such, finding a minimum weight error vector \mathbf{e} is equivalent to minimizing the Hamming distance

$$\min_{\mathbf{z}} d(\mathbf{x}, \mathbf{G}\mathbf{z}).$$

An effective algorithm for this purpose has emerged via survey propagation [16] and its variants [18]–[21], [23], which share a message-passing nature with belief propagation, but require in general many runs with pruning stages between runs in order to converge. This results in a higher computational complexity. A simpler alternative would thus be of interest.

III. BELIEF PROPAGATION

We review the standard belief propagation algorithm here to facilitate the modified algorithm of Section IV. Consider the factor graph of the generator equation $\mathbf{x} = \mathbf{G}\mathbf{z}$, as in Fig. 1. If \mathbf{x} is not in the column space of the generator matrix \mathbf{G} , this gives an inconsistent system; belief propagation attempts to iteratively refine the information bits in \mathbf{z} so as to reconcile $\mathbf{G}\mathbf{z}$ with \mathbf{x} . The algorithm passes messages along edges in the graph between variable nodes (denoted as circles) and parity-check nodes (denoted as squares) [29]. These messages consist of two-element vectors containing pseudo-probabilities that sum to one.

The update equations at the variable nodes appear as

$$m_{z_i \rightarrow f_j}(z_i) = \zeta \Pr(z_i) \prod_{\substack{k \in F(i) \\ k \neq j}} m_{f_k \rightarrow z_i}(z_i) \quad (3)$$

where $m_{f_k \rightarrow z_i}(z_i)$ denotes an incoming messages at variable node i , $F(i)$ contains the indices k whose parity-check nodes f_k connect to variable node z_i , and $m_{z_i \rightarrow f_j}(z_i)$ denotes an outgoing message sent from variable node z_i to parity-check node f_j . The scale factor ζ (for a given node) is chosen to ensure that evaluations sum to one:

$$m_{z_i \rightarrow f_j}(0) + m_{z_i \rightarrow f_j}(1) = 1.$$

The variable node probability $\Pr(z_i)$ reflects any *a priori* information on the information bit z_i , if available.

The update equations at the parity check nodes read as

$$m_{f_j \rightarrow z_i}(z_i) = \frac{1}{2} \left\{ 1 + (-1)^{z_i} (1 - 2m_{x_j \rightarrow f_j}(1)) \right. \\ \left. \times \prod_{\substack{k \in V(j) \\ k \neq i}} (1 - 2m_{z_k \rightarrow f_j}(1)) \right\} \quad (4)$$

where $V(j)$ contains the indices k of the variable nodes z_k which enter into the j -th parity check. The “flooding schedule” runs (3) at each variable node, followed by (4) at each parity-check node, in successive iterations. If convergence occurs, the belief values for the bits $\{z_i\}$ are given as

$$b_i(z_i) = \zeta \Pr(z_i) \prod_{k \in F(i)} m_{f_k \rightarrow z_i}(z_i),$$

where, again, ζ ensures that evaluations $b_1(0)$ and $b_i(1)$ sum to one.

Let $\hat{x}_i \in \{0, 1\}$ denote the candidate i -th codeword bit from the generator matrix, as distinguished from the actual bit x_i the algorithm is attempting to match. By conventional belief propagation, the message $m_{x_j \rightarrow f_j}(\hat{x}_j)$ is that sent from the parity-check node $q_j(\hat{x}_j)$ (right-most squares in 1) to the variable node x_j . Setting $q_j(1)$ to some “soft” probability skewed inward from the hard value x_j gives an algorithm which, in general, converges to seemingly meaningless probabilities. Setting $q_j(1) = x_j$ (hard values) gives, in general, an inconsistent system, resulting in numerical singularities.

IV. TRUTHINESS PROPAGATION

To overcome the shortcomings of standard belief propagation, consider the choice

$$q_j(1) = \alpha x_j + (1 - \alpha) m_{x_j \rightarrow q_j}(1) \quad (5)$$

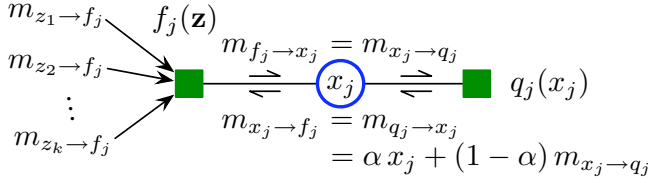


Fig. 2. Messages at j -th parity-check node $f_j(\mathbf{z})$, to illustrate modified algorithm. The constraint node q_j sends back a convex combination of its hard constraint x_j and its incoming message $m_{x_j \rightarrow q_j}$, controlled by a “truthiness” factor α in (5).

TABLE I

SUMMARY OF TRUTHINESS PROPAGATION ALGORITHM; SUPERSCRIPIT (n) DENOTES AN ITERATION COUNTER

Initialization: $m_{z_i \rightarrow f_j}^{(0)}(1) = 0.5 \pm \text{dither}$

Natural parities:

$$m_{f_j \rightarrow x_j}^{(n)}(1) = \frac{1}{2} \left\{ 1 - \prod_{k \in V(j)} (1 - 2m_{z_k \rightarrow f_j}^{(n)}(1)) \right\}$$

$$m_{x_j \rightarrow f_j}^{(n)}(1) = \alpha x_j + (1 - \alpha) m_{f_j \rightarrow x_j}^{(n)}(1)$$

Check nodes:

$$m_{f_j \rightarrow z_i}^{(n)}(z_i) = \frac{1}{2} \left\{ 1 + (-1)^{z_i} (1 - 2m_{x_j \rightarrow f_j}^{(n)}(1)) \right. \\ \left. \times \prod_{\substack{k \in V(j) \\ k \neq i}} (1 - 2m_{z_k \rightarrow f_j}^{(n)}(1)) \right\}$$

Variable nodes:

$$m_{z_i \rightarrow f_j}^{(n+1)}(z_i) = \zeta_{ij} \prod_{\substack{k \in F(i) \\ k \neq j}} m_{f_k \rightarrow z_i}^{(n)}(z_i)$$

Beliefs:

$$b_i^{(n+1)}(z_i) = \zeta_i \prod_{k \in F(i)} m_{f_k \rightarrow z_i}^{(n)}(z_i)$$

where $0 < \alpha < 1$ and

$$m_{x_j \rightarrow q_j}(\hat{x}_j) = m_{f_j \rightarrow x_j}(\hat{x}_j) \\ = \frac{1}{2} \left\{ 1 + (-1)^{\hat{x}_j} \prod_{k \in V(j)} (1 - 2m_{z_k \rightarrow f_j}^{(n)}(1)) \right\} \quad (6)$$

is the message sent from parity-check node f_j to variable node x_j ; cf. Fig. 2. The value $m_{f_j \rightarrow x_j}(1)$ is the probability that the information bits $\{z_k\}$ which impinge on the j -th parity check produce an odd parity [30], giving thus the “natural parity” (i.e., without the constraint from x_j) at that check node.

The rationale for choosing the convex combination in (5) is that if the natural parity agrees with the constraint x_j , the constraining node (labeled q_j in Fig. 2) maintains this “hard” constraint. If instead the natural parity is opposite to the constraint x_j , the probability fed from q_j is softened, thus allowing further refinement in the message passing algorithm. The operation at the constraining node is akin to a system which feeds back not the actual “truth” (as strict belief propagation would do), but rather what it “wishes” to be true, via injection of the natural parity. Accordingly, we dub the al-

TABLE II
SUM-PRODUCT FORM OF THE T-A-P ALGORITHM FROM [25];
SUPERSCRIPIT (n) DENOTES AN ITERATION COUNTER

Initialization:

$$m_{z_i \rightarrow f_j}^{(0)}(1) = 0.5 \pm \text{dither}$$

$$m_{x_j \rightarrow f_j}(1) = \exp[\beta(2x_j - 1)] / [\exp(\beta) + \exp(-\beta)]$$

Check nodes:

$$m_{f_j \rightarrow z_i}^{(n)}(z_i) = \frac{1}{2} \left\{ 1 + (-1)^{z_i} (1 - 2m_{x_j \rightarrow f_j}(1)) \right. \\ \left. \times \prod_{\substack{k \in V(j) \\ k \neq i}} (1 - 2m_{z_k \rightarrow f_j}^{(n)}(1)) \right\}$$

Softened beliefs:

$$c_i^{(n)}(z_i) = \gamma b_i^{(n)}(z_i) + \frac{(1 - \gamma)}{2}$$

Variable nodes:

$$m_{z_i \rightarrow f_j}^{(n+1)}(z_i) = \zeta_{ij} c_i^{(n)}(z_i) \prod_{\substack{k \in F(i) \\ k \neq j}} m_{f_k \rightarrow z_i}^{(n)}(z_i)$$

Updated beliefs:

$$b_i^{(n+1)}(z_i) = \zeta_i c_i^{(n)}(z_i) \prod_{k \in F(i)} m_{f_k \rightarrow z_i}^{(n)}(z_i)$$

gorithm *truthiness propagation*¹, obtained by running (3), (6), (5) [giving $m_{x_j \rightarrow f_j}(1) = q_j(1)$], and then (4), and iterating; the resulting algorithm is summarized in Table I. Interestingly, this simple modification gives an algorithm that is observed to converge to a quantizing solution (the beliefs $b_i(z_i)$ tend to 0 or 1), yet features negligible overhead compared to standard belief propagation. It is thus an order of magnitude simpler than the survey propagation methods developed in [16], [18]–[21], or the pruning approach applied to belief propagation [24].

V. SIMULATION RESULTS

Performance is compared with the Thouless-Anderson-Palmer approach of Murayama [31], [25], which is likewise a message passing algorithm derived from the factor graph of a low density generator matrix. Although the algorithm as published in [25] requires hyperbolic tangent and hyperbolic arctangent operations (presenting a significantly higher computational complexity per iteration), some straightforward algebraic manipulations can transform the algorithm into the computationally simpler sum-product form summarized in Table II. In this algorithm, the parity-check nodes integrate soft probabilities derived from $\{x_i\}$, as controlled by a parameter β in the notations of [31], [25], while the variable nodes replace the prior probabilities $\Pr(z_i)$ by softened beliefs from the previous iteration, as controlled by a reinjection parameter γ . As with the parameter α of the proposed algorithm, analytic formulas for the optimal choices of β and γ are not available at

¹**truthiness** (noun): “The quality of preferring concepts or facts one wishes to be true, rather than concepts or facts known to be true” (Merriam-Webster).

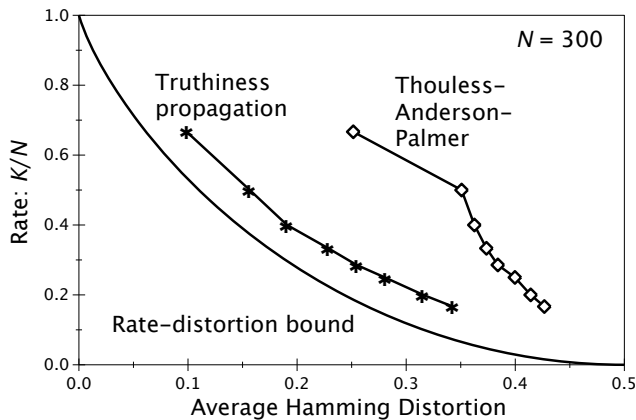


Fig. 3. Comparison of truthiness propagation with T-A-P [25] for short block lengths ($N = 300$) and regular generator matrices.

present, but are instead adjusted manually.² The survey propagation algorithm of [19], [18], finally, requires even higher computational complexity per iteration, additional parameters requiring manual adjustment (denoted w_{sou} and w_{info} in [19]), and subsequent pruning steps between runs, as with [24]. These pruning-based approaches give a more cumbersome procedure outright, and although the reported quantization performance in [19] and [24] is close to the rate-distortion curve, improvements over [25] are hardly discernable, and so the pruning algorithms are not included in the comparisons to follow.

Regular low density generator matrices were initially used, as in [25], featuring two ones per row and C ones per column, giving an adjustable code rate of $R = 2/C$. Using a block length of 300 bits, the obtained rate-distortion performance is plotted in Fig. 3. The average distortion at a given rate is

$$D = \frac{E[d(\mathbf{x}, \mathbf{Gz})]}{N}$$

in which the expectation is replaced by the empirical mean of 20,000 quantization runs (40 different generator matrices, with 500 independent runs each), using identical data for the two algorithms at each run. (The conventional belief propagation algorithm was also run, but gave an average distortion of about 0.5 at each rate, which is no better than a coin toss.)

The seemingly poor performance of the T-A-P algorithm is due to the short block length. Fig. 4 shows a histogram of the Hamming distortion rate over the 20,000 runs for the two algorithms at rate $R = 1/2$, illustrating a “double-hump” character for the T-A-P algorithm. (Similar histograms were observed for all code rates). The left-most hump shows excellent quantization performance; the right-most hump reflects the sizeable number of runs for which the beliefs failed to converge to a meaningful solution, even after 1000 iterations of the message passing algorithm. The truthiness propagation algorithm, by contrast, shows a single hump,

²Simulations indicate that reducing α or β generally gives lower quantization distortion, until a lower limit on either is reached, below which the respective algorithms fail to converge. The critical lower value for α or β decreases with the code rate; empirically, the lower bound $\alpha \geq 1 - 2D_{\min}(K/N)$ with D_{\min} the minimum Hamming distortion at rate K/N [cf. (2)], was found to work well. Some indications on choosing β are suggested from [31]. As in [25], $\gamma = 0.1$ was used.

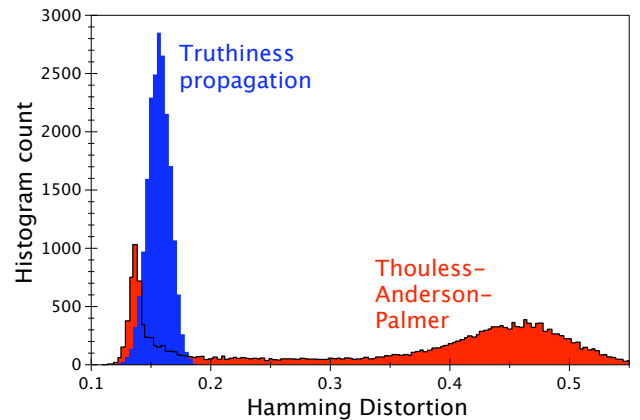


Fig. 4. Histogram of Hamming distortion over 20,000 runs for the two algorithms, at rate $R = 1/2$ and with block length $N = 300$.

exhibiting low variance about the mean value. In addition, whereas the convergent runs of the T-A-P algorithm required typically hundreds of iterations for the belief values to settle, the truthiness propagation algorithm was observed to converge within tens of iterations in most runs.

Simulations indicate that if the block length is increased into the tens of thousands, the left-most hump for the T-A-P scheme dominates the histogram; the performance then concurs with the respectable results reported in [25]. This indicates that the convergence difficulties noted here are likely due to a factor graph girth deficiency which becomes more prominent in the short block length case. Interestingly, the truthiness propagation algorithm does not appear to suffer these convergence difficulties, despite using the same generator matrices.

Further performance improvements were observed using irregular generator matrices, with the results plotted in Fig. 5; the degree distribution polynomials (in the formulation of [32]) for the generator matrices are listed in Table III for code rates up to $R = 0.6$; codes of higher rate were obtained as the duals of lower rate codes from Table III. The T-A-P scheme, unfortunately, failed to converge for any generator matrices featuring check node degrees exceeding two, indicating that the formulation is apparently functional only for the “pairwise interaction” model. The conventional belief propagation algorithm *did* converge for specific code rates using the irregular generator matrices, although the quantization performance, as plotted in Fig. 5, is less than spectacular.

VI. CONCLUDING REMARKS

We have proposed a simple modification to the standard belief propagation algorithm, which is observed to yield a convergent algorithm for the code word quantization problem. The algorithm is much simpler in implementation and complexity compared to survey propagation [16], [18]–[21], and avoids the cumbersome pruning and decimation steps required of that procedure and others [24]. Its main advantage concerns its consistent performance in the short block-length case. For longer block lengths, the quantization results reported in [25], [19], [21] inch slightly closer to the rate-distortion curve. For such cases, the algorithm of [25] (once transcribed to its sum-product form of Table II) is perhaps preferred, in view of its

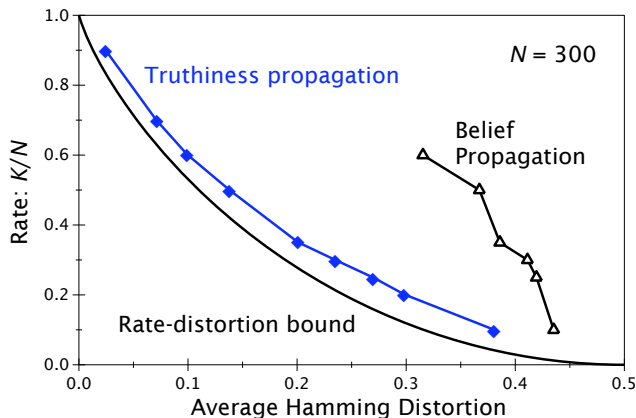


Fig. 5. Rate-distortion performance for the truthiness propagation algorithm using irregular low density generator matrices, for the short block-length case.

TABLE III

COEFFICIENTS OF DEGREE DISTRIBUTION POLYNOMIALS $\rho(\zeta) = \sum_{i \geq 2} \rho_i \zeta^{i-1}$ AND $\lambda(\zeta) = \sum_{i \geq 2} \lambda_i \zeta^{i-1}$ FOR THE IRREGULAR GENERATOR MATRICES

	rate = 0.1	0.2	0.25	0.3	0.35	0.5	0.6
ρ_2	0.125611	0.081547	0.106098	0.132994	0.157703	0.344865	0.303078
ρ_3	0.216442	0.198215	0.161547	0.125532	0.199106	0.255370	0.145294
ρ_4							0.003672
ρ_5				0.163834		0.431081	0.164982
ρ_6			0.017169	0.148921	0.032484		
ρ_7	0.077057		0.240904	0.198548			
ρ_8	0.214279				0.080705		0.037698
ρ_9	0.001705						0.032870
ρ_{10}			0.000245				
ρ_{13}	0.001975						0.137459
ρ_{20}		0.720238		0.428719			
ρ_{22}					0.299788		
ρ_{23}					0.031081		
ρ_{30}	0.362930		0.474037				
ρ_{34}					0.000585		0.136910
ρ_{35}							0.038038
λ_6						1	0.5
λ_7							0.5
λ_{14}					1		
λ_{17}				0.2			
λ_{18}				0.8			
λ_{25}			1				
λ_{35}		1					
λ_{54}	1						

simple computational requirements and excellent performance, at least for pairwise interaction models.

REFERENCES

- [1] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384–386, May 1978.
- [2] T. Johansson and F. Jönsson, "On the complexity of some cryptographic problems based on the general decoding problem," *IEEE Trans. Inf. Theory*, vol. 48, no. 10, pp. 2669–2678, Oct. 2002.
- [3] V. Guruswami and A. Vardy, "Maximum-likelihood decoding of Reed–Solomon codes is NP-hard," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2249–2256, July 2005.
- [4] W. Trappe and L. C. Washington, *Introduction to Cryptography with Coding Theory*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2006.
- [5] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications*, G. Cohen and J. Wolfman, eds., vol. 338, pp. 106–113. Berlin: Springer, 1989.
- [6] A. Canteaut and F. Chabaud, "A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 367–378, Jan. 1998.
- [7] C.-C. Shih, C. R. Wulff, C. R. P. Hartmann, and C. K. Mohan, "Efficient heuristic search algorithms for soft-decision decoding of linear block codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 3023–3038, Nov. 1998.
- [8] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 439–441, May 1983.
- [9] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 563–593, Mar. 2003.
- [10] R. J. Barron, B. Chen, and G. W. Wornell, "The duality between information embedding and source coding with side information and some applications," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1159–1180, May 2003.
- [11] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1181–1203, May 2003.
- [12] J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Process.*, vol. 10, no. 53, pp. 3923–3935, Oct. 2005.
- [13] J. Fridrich, M. Goljan, and D. Soukal, "Wet paper codes with improved coding efficiency," *IEEE Trans. Inf. Forens. Security*, vol. 1, no. 1, pp. 102–110, Mar. 2006.
- [14] P. Regalia, "Cryptographic secrecy of steganographic matrix embedding," *IEEE Trans. Inf. Forens. Security*, vol. 3, no. 4, pp. 786–791, Dec. 2008.
- [15] A. Khisti, E. Martinian, and G. W. Wornell, "Information embedding with distortion side information," in *Proc. Int. Symp. Inform. Theory*, July 2006, pp. 183–187.
- [16] M. Mézard and R. Zecchina, "Random K -satisfiability problem: from an analytic solution to an efficient algorithm," *Physical Review E*, vol. 66, no. 5, 2002.
- [17] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 10, pp. 1491–1513, Oct. 2002.
- [18] E. Martinian and J. S. Yedidia, "Iterative quantization using codes on graphs," in *Proc. Allerton Conf. Commun., Control Computing*, Oct. 2003.
- [19] M. J. Wainwright and E. Maneva, "Lossy source coding via message-passing and decimation over generalized codewords of LDGM codes," in *Proc. Int. Symp. Inform. Theory*, Sep. 2005.
- [20] E. Martinian and M. J. Wainwright, "Low-density constructions for lossy compression, binning, and coding with side information," in *Proc. IEEE Inform. Theory Workshop*, 2006, pp. 263–264.
- [21] M. J. Wainwright, "Sparse graph codes for side information and binning," *IEEE Signal Process. Mag.*, vol. 24, no. 7, pp. 47–57, Sep. 2007.
- [22] A. Gupta and S. Verdú, "Nonlinear sparse-graph codes for lossy compression of discrete nonredundant sources," in *Proc. Inform. Theory Workshop*, Sep. 2007, pp. 541–546.
- [23] R. Tu, Y. Mao, and J. Zhao, "On generalized survy propagation: normal realizations and sum-product interpretation," in *Proc. Int. Symp. Inform. Theory*, July 2006, pp. 2042–2046.
- [24] T. Filler and J. Fridrich, "Binary quantization using belief propagation with decimation over factor graphs of LDGM codes," in *Proc. Allerton Conf. Commun., Control Computing*, 2007.
- [25] T. Murayama, "Thouless-Anderson-Palmer approach for lossy compression," *Physical Review E*, vol. 69, 2004.
- [26] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ: Wiley, 2007.
- [27] E. Martinian and M. J. Wainwright, "Low-density constructions can achieve the Wyner-Ziv and Gelfand-Pinsker bounds," in *Proc. Int. Symp. Inform. Theory*, July 2006, pp. 484–488.
- [28] E. Martinian and M. J. Wainwright, "Low density codes achieve the rate-distortion bound," in *Proc. Data Compression Conf.*, Mar. 2006.
- [29] 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.
- [30] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 2, pp. 21–28, 1962.
- [31] T. Murayama and M. Okada, "One step RSB scheme for the rate distortion function," *J. Physics A: Mathematical General*, vol. 36, pp. 11123–11130, 2003.
- [32] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.