On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit

Sae-Young Chung, *Member, IEEE*, G. David Forney, Jr., *Fellow, IEEE*, Thomas J. Richardson, and Rüdiger Urbanke

***Abstract*—We develop improved algorithms to construct good low-density parity-check codes that approach the Shannon limit very closely. For rate 1/2, the best code found has a threshold within 0.0045 dB of the Shannon limit of the binary-input additive white Gaussian noise channel. Simulation results with a somewhat simpler code show that we can achieve within 0.04 dB of the Shannon limit at a bit error rate of 10^{-6} using a block length of 10^7.**

***Index Terms*—Density evolution, low-density parity-check codes, Shannon limit, sum-product algorithm.**

I. Introduction

RICHARDSON *et al.* [1] constructed irregular low-density parity-check (LDPC) codes that easily beat the best known turbo codes when the block length of the code is large. This shows that LDPC codes—originally invented by Gallager [2], forgotten for decades, rediscovered by many [3], [4]—can now outperform powerful turbo codes [5] if designed properly.

They also showed that for many interesting channels, including additive white Gaussian noise (AWGN) channels, one can use an algorithm called *density evolution* [6] to calculate a threshold value for a randomly constructed irregular LDPC code which determines the boundary of the error-free region asymptotically as the block length tends to infinity. (Density evolution based on combinatorial or Monte Carlo approaches had been previously attempted by Gallager [2], Spielman [7], and MacKay [8].)

In this letter, we develop an improved implementation of density evolution called *discretized density evolution* [9]. We show that this improved algorithm models the exact behavior of discretized sum-product decoding. Using this algorithm and an improved optimization algorithm, we design good rate-1/2 irregular LDPC codes for binary-input AWGN channels that approach the Shannon limit very closely.

II. Discretized Density Evolution

Our derivation of discretized density evolution is based on the "local tree assumption" of [1], which is validated by the general concentration theorem of [1].

Manuscript received July 20, 2000. The associate editor coordinating the review of this letter and approving it for publication was Dr. M. Fossorier.

S.-Y. Chung was with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is now with Airvana, Inc., Chelmsford, MA 01824 USA (e-mail: sae-young.chung@airvananet.com).

G. D. Forney, Jr. is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: forney@lids.mit.edu).

T. J. Richardson is with Flarion Technologies, Bedminster, NJ 07921 USA (e-mail: richardson@flarion.com).

R. Urbanke is with the Communications Theory Lab, EPFL, Lausanne, Switzerland (e-mail: Rudiger.Urbanke@epfl.ch).

Publisher Item Identifier S 1089-7798(01)02122-6.

Let v be a log-likelihood ratio (LLR) message from a degree-d_v variable node to a check node. Under sum-product decoding, v is equal to the sum of all incoming LLRs; i.e.,

$$v = \sum_{i=0}^{d_v-1} u_i \tag{1}$$

where u_i, $i = 1, \ldots, d_v - 1$ are the incoming LLR's from the neighbors of the variable node except the check node that gets the message v, and u_0 is the observed LLR of the output bit associated with the variable node.

The message update rule for check nodes can be obtained by observing the duality between variable and check nodes and the resulting Fourier transform relationship [10]. From this, we get the following well known "tanh rule" (see [11]):

$$\tanh\frac{u}{2} = \prod_{j=1}^{d_c-1} \tanh\frac{v_j}{2} \tag{2}$$

where v_j, $j = 1, \ldots, d_c - 1$ are the incoming LLR's from $d_c - 1$ neighbors of a degree-d_c check node, and u is the output LLR message sent to the remaining neighbor.

From now on, we assume a random ensemble of irregular codes specified by two degree distributions $\lambda(x)$ and $\rho(x)$, where $\lambda(x) = \sum_{i=2}^{d_l} \lambda_i x^{i-1}$ and $\rho(x) = \sum_{j=2}^{d_r} \rho_j x^{j-1}$. Here, λ_i is the fraction of edges that belong to degree-i variable nodes, ρ_j is the fraction of edges that belong to degree-j check nodes, d_l is the maximum variable degree, and d_r is the maximum check degree.

Let $\mathcal{Q}(w)$ be the quantized message of w, i.e.,

$$\mathcal{Q}(w) = \begin{cases} \left\lfloor \dfrac{w}{\Delta} + \dfrac{1}{2} \right\rfloor \cdot \Delta, & \text{if } w \geq \dfrac{\Delta}{2} \\ \left\lceil \dfrac{w}{\Delta} - \dfrac{1}{2} \right\rceil \cdot \Delta, & \text{if } w \leq -\dfrac{\Delta}{2} \\ 0, & \text{otherwise} \end{cases}$$

where $\mathcal{Q}$ is the quantization operator; Δ is the quantization interval; $\lfloor x \rfloor$ is the largest integer not greater than x; and $\lceil x \rceil$ is the smallest integer not less than x.

Discretized sum-product decoding is defined as sum-product decoding with all input and output messages quantized in this way. Under discretized sum-product decoding, (1) becomes $\overline{v} = \sum_{i=0}^{d_v-1} \overline{u}_i$, where $\overline{v} = \mathcal{Q}(v)$ and $\overline{u}_i = \mathcal{Q}(u_i)$ for $i = 0, \ldots, d_v - 1$. We denote the probability mass function

(pmf) of a quantized message $\overline{w}$ by $p_w[k] = \Pr(\overline{w} = k\Delta)$ for $k \in \mathbb{Z}$. Then, p_v is related to its input pmf's by

$$p_v = \bigotimes_{i=0}^{d_v-1} p_{u_i}$$

where p_v is the pmf of $\overline{v}$; p_{u_i} is the pmf of $\overline{u}_i$; and $\bigotimes$ is discrete convolution. Since the $\overline{u}_i$'s are independent and identically distributed (i.i.d.) for $1 \le i < d_v$, the above can be rewritten as

$$p_v = p_{u_0} * \left(\bigotimes^{d_v-1} p_u \right)$$

where $p_u = p_{u_i}$, $1 \le i < d_v$ and $*$ is discrete convolution. This calculation can be done efficiently using an FFT.

We define the following two-input operator $\mathcal{R}$:

$$\mathcal{R}(a, b) = \mathcal{Q}\left(2\tanh^{-1}\left(\tanh\frac{a}{2}\tanh\frac{b}{2}\right)\right)$$

where a and b are quantized messages. Note that this operation can be done using a pre-computed table, which is the key step for making discretized density evolution computationally efficient. Using this operator, we calculate the quantized message $\overline{u}$ of (2) as follows:

$$\overline{u} = \mathcal{R}(\overline{v}_1, \mathcal{R}(\overline{v}_2, \ldots, \mathcal{R}(\overline{v}_{d_c-2}, \overline{v}_{d_c-1}) \ldots))$$

where we assume that discretized sum-product decoding at check nodes is done pairwise.

Let $c = \mathcal{R}(a, b)$. The pmf p_c of c is given by

$$p_c[k] = \sum_{(i,j):\, k\Delta = \mathcal{R}(i\Delta, j\Delta)} p_a[i] p_b[j].$$

Abusing notation, we write this as $p_c = \mathcal{R}(p_a, p_b)$.

Since the p_{v_i}'s are all equal, we define $p_v = p_{v_i}$ for any $1 \le i < d_c$, and write $p_u = \mathcal{R}(p_v, \mathcal{R}(p_v, \ldots, \mathcal{R}(p_v, p_v), \ldots))$ as $p_u = \mathcal{R}^{d_c-1} p_v$. Note that we can reduce the number of operations to do this by properly nesting this calculation.[1] By defining $\lambda(p) = \sum_{i=2}^{d_l} \lambda_i \bigotimes^{i-1} p$ and $\rho(p) = \sum_{j=2}^{d_r} \rho_j \mathcal{R}^{j-1} p$ for any pmf p, we can write discretized density evolution as follows:

Theorem 1: Discretized density evolution is described by

$$p_u^{(\ell+1)} = \rho(p_{u_0} * \lambda(p_u^{(\ell)}))$$

where the initial pmf $p_u^{(0)}$ has all mass at 0 and ℓ is the iteration number.

To run this algorithm, without loss of generality, we first assume that the all-0 codeword was sent. Then, we fix the channel parameter, namely noise power, and we run the above algorithm iteratively until either the density of u tends to the "point mass at infinity" (equivalently, the probability of error tends to zero), or it converges to a density having a finite probability of error, which is defined as the probability of u being negative. The threshold is defined as the maximum noise level such that the probability of error tends to zero as the number of iterations tends to infinity.

The complexity of this calculation is of order $O(n^2)$ due to the calculations at check nodes, where n is the number of quantization levels. However, this is actually faster than the calculation based on changing of measures between LLR and $(p_0 - p_1)$ domains as in [1], which requires finer quantization due to numerical problems when changing measures. As a result, our algorithm is more accurate and also realizes the discretized version of the sum-product algorithm exactly. This implies that the threshold predicted by our algorithm is always a lower bound, since it is exact for a sub-optimal decoding.

Table I shows how the threshold values are affected by quantization for the rate-1/2 $d_l = 200$ code in Table II. Number of bits used for quantization versus threshold values in σ (noise standard deviation) and errors in dB relative to the threshold value for 14-bit quantization are shown. All threshold values are rounded down. When 14 bits are used to quantize the LLR values into 16 384 levels, we observe that the threshold has 6-digit precision.

TABLE I
QUANTIZATION EFFECT

bits	threshold (σ)	error [dB]
9	0.975122	0.01708
10	0.976918	0.00109
11	0.977025	0.00014
12	0.977037	0.00003
13	0.977040	0.00001
14	0.977041	0.00000

TABLE II
GOOD RATE-1/2 CODES WITH $d_l = 100, 200, 8000$

d_l	100		200		8000	
	x	λ_x	x	λ_x	x	λ_x
	2	0.170031	2	0.153425	2	0.096294
	3	0.160460	3	0.147526	3	0.095393
	6	0.112837	6	0.041539	6	0.033599
	7	0.047489	7	0.147551	7	0.091918
	10	0.011481	18	0.047938	15	0.031642
	11	0.091537	19	0.119555	20	0.086563
	26	0.152978	55	0.036379	50	0.093896
	27	0.036131	56	0.126714	70	0.006035
	100	0.217056	200	0.179373	100	0.018375
					150	0.086919
					400	0.089018
					900	0.057176
					2000	0.085816
					3000	0.006163
					6000	0.003028
					8000	0.118165
ρ_{av}	10.9375		12.0000		18.5000	
σ	0.97592		0.97704		0.9781869	
SNR_{norm}	0.0247		0.0147		0.00450	

III. OPTIMIZATION

In this letter, we use a slightly different optimization technique than the iterative linear programming used in [1] to opti-

[1]Although for finite quantization different nesting might produce slightly different results, it still corresponds to some valid decoding algorithm and asymptotically, as $\Delta \to 0$, it will converge to continuous belief propagation.

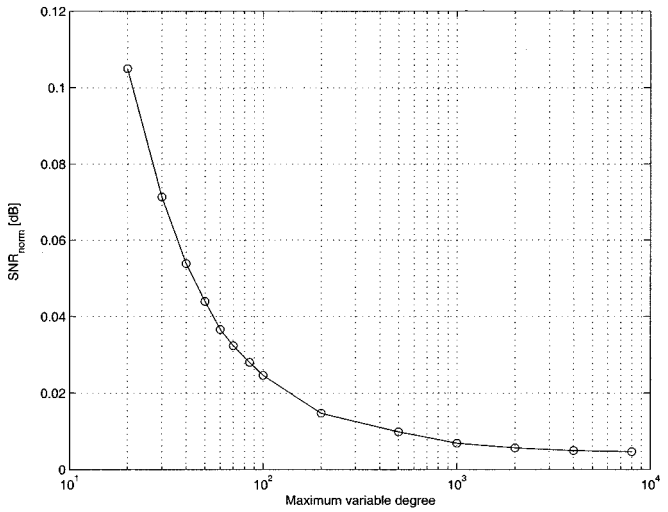

Fig. 1. Threshold ($\mathrm{SNR}_{\mathrm{norm}}$) of rate-1/2 LDPC codes with maximum variable degree = $20, \ldots, 8000$.

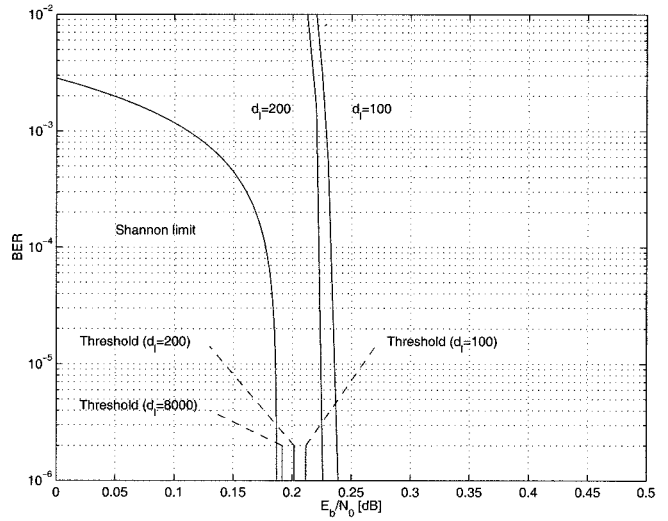

Fig. 2. Simulation results for the $d_l = 100$ and $d_l = 200$ codes of Table II, using a block length of 10^7.

mize $\lambda(x)$, which seems to be better for designing codes, especially when d_l is large. We maximize the rate of the code while maintaining the following constraints: 1) $\lambda(1) = 1$; 2) the new $\lambda(x)$ is not significantly different from the old one (required to guarantee that the linear programming formulation is valid); and 3) the new $\lambda(x)$ is better than the old one (produces smaller probability of error). For details of this algorithm, see [9].

Thresholds for some good rate-1/2 codes with maximum variable degrees 20, 30, 40, 50, 60, 70, 85, 100, 200, 500, 1000, 2000, 4000, and 8000 are given in Fig. 1 [9], where $\mathrm{SNR}_{\mathrm{norm}}$ is defined as the distance from the Shannon limit in dB. Table II gives the degree distributions for some of these codes, where threshold values are rounded down and $\mathrm{SNR}_{\mathrm{norm}}$ values are rounded up. The Shannon limit is $\sigma = 0.978\,69$ at rate 1/2.

We used concentrated $\rho(x)$'s only, where $\rho(x) = (1-\rho)x^{j-1} + \rho x^j$ for some integer $j \geq 2$. This restriction not only makes it easier to optimize $\rho(x)$, especially for large maximum variable degrees, but also is not too restrictive for the AWGN channel [12]. The average check degree ρ_{av} is used in Table II to parametrize $\rho(x)$ where $\rho_{\mathrm{av}} = (1-\rho)(j-1) + \rho j = j - 1 + \rho$.

In Fig. 2, we show simulation results for the $d_l = 100, 200$ codes in Table II. A block length of 10^7 was used, and the code graph was constructed randomly, except that cycles of length 2 and 4 were avoided. At BER = 10^{-6}, the $d_l = 200$ code operates within 0.04 dB of the Shannon limit. The maximum allowed number of iterations was 2000. When decoding was successful, about 800–1100 iterations were needed. No undetected errors occurred.

At a BER of about 10^{-6}, we decoded 249 and 371 blocks for the $d_l = 100$ and $d_l = 200$ codes, respectively—i.e., about 1.2×10^9 and 1.8×10^9 decoded bits, respectively.

More results and an online demonstration of density evolution are available at http://truth.mit.edu/~sychung.

IV. Conclusions

We have designed LDPC codes for binary-input AWGN channels that approach the Shannon limit very closely. Our design is based on discretized density evolution and optimization of degree distributions via iterative linear programming. Our design can be used to design LDPC codes of arbitrary rates between 0 and 1 for many interesting channels.

Our results strongly suggest that optimal LDPC codes under sum-product decoding for AWGN channels may approach the Shannon limit asymptotically as the block length tends to infinity.

Acknowledgment

This work is based on S.-Y. Chung's Ph.D. dissertation.

References

[1] T. J. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, Feb. 2001.

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

[3] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710–1722, Nov. 1996.

[4] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[6] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, Feb. 2001.

[7] D. A. Spielman, "Finding good LDPC codes," in *Proc. 36th Allerton Conf. Commun. Contr., and Comput.*, Sept. 1998.

[8] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[9] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 2000.

[10] G. D. Forney Jr., "Codes on graphs: Normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, Feb. 2001.

[11] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[12] S.-Y. Chung, T. J. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, Feb. 2001.