Optimized Low-Density Parity-Check Codes for Partial Response Channels

Nedeljko Varnica and Aleksandar Kavčić

Abstract— **We optimize irregular low-density parity-check (LDPC) codes to closely approach the i.u.d. capacities of partial response channels. In our approach we use the degree sequences optimization method for memoryless channels proposed by Richardson, Shokrollahi and Urbanke and appropriately modify it for channels with memory. With this optimization algorithm we construct codes whose noise tolerance thresholds are within 0.15dB of the i.u.d. channel capacities. Our simulation results show that irregular LDPC codes with block lengths 10^6 bits yield bit error rates 10^{-6} at signal-to-noise ratios 0.22dB away from the channel capacities.**

Keywords— **Low-density parity-check codes, density evolution, i.u.d. channel capacity, partial response channels.**

I. Introduction

SINCE the invention of turbo codes [1] and the rediscovery of low-density parity-check (LDPC) codes [2] (originally invented by Gallager [3]) the Shannon limits for several memoryless channel models, such as the erasure channels and the memoryless binary-input additive white Gaussian noise (AWGN) channels, have been practically achieved. The design of capacity achieving irregular LDPC codes is based on the optimization of the edge degree sequences in a bipartite LDPC code graph. Luby et al. originally proposed this optimization for the binary erasure channels [4]. This concept was broadened to a variety of channels, including the memoryless AWGN channels [5][6]. However, the capacity achieving code constructions for partial response channels remained an open problem.

The i.u.d. capacity is the information rate when the channel inputs are independent and uniformly distributed (i.u.d.). From [7] we know that random linear codes (which are impractical) can achieve the i.u.d. capacities of the partial response channels. However, it is not clear whether LDPC codes can approach the i.u.d. capacities of these channels. In this letter we construct irregular LDPC codes with noise tolerance thresholds [5] within 0.15dB of the i.u.d. channel capacities. Our code design strategy is to optimize the LDPC code degree sequences to achieve higher noise tolerance thresholds. The optimization is performed using the density evolution algorithm for channels with intersymbol interference (ISI) memory [7].

The described optimization yields codes that exhibit a performance very near the i.u.d. capacity. Our simulation results for codes of length 10^6 bits show that bit error rates 10^{-6} are achieved at signal-to-noise ratios (SNRs) less than 0.22dB away from the i.u.d. capacity limits.

The authors are with the Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: varnica@hrl.harvard.edu, kavcic@hrl.harvard.edu)

This work was supported by the National Science Foundation under Grant No. CCR-0118701

II. Background

A partial response channel is a binary-input/continuous-output AWGN channel with ISI memory. The probabilistic law of this channel is captured by

$$Y_t = \sum_{k=0}^{\nu} h_k X_{t-k} + W_t \ , \qquad (1)$$

where ν is the length of the channel memory. $W_1^N = [W_1, W_2, ..., W_N]$ represents a zero-mean white Gaussian random vector with $W_t \sim \mathcal{N}(0, \sigma^2)$. X_1^N denotes the antipodal binary input vector and Y_1^N is the discrete-time continuous-valued output vector. The channel memory is captured by the vector of coefficients $\underline{h} = [h_0, h_1, ..., h_\nu]$, often given in polynomial form $h(D) = \sum_{i=0}^{\nu} h_i D^i$.

The i.u.d. capacity of a channel is defined as

$$C_{\text{i.u.d.}} = \lim_{N\to\infty} \frac{1}{N} \cdot \mathrm{I}\left(\mathrm{X}_1^{\mathrm{N}}; \mathrm{Y}_1^{\mathrm{N}}\right)|_{\mathrm{P}_{\mathrm{X}_1^{\mathrm{N}}}(\mathrm{x}_1^{\mathrm{N}})=2^{-\mathrm{N}}} \ , \qquad (2)$$

i.e., it is the information rate when the input sequence X_1^N is i.u.d. A numerical method for computing this capacity for channels with ISI memory has recently been proposed in [8] and [9]. This capacity represents an upper bound on the achievable rates (such that the probability of error can be made arbitrarily small) for random linear binary codes for channels with memory [7].

Our goal is to design LDPC codes that approach this capacity. In the sequel we use the normal code/channel graph introduced in [7] since we seek to design codes for partial response channels. Following [4] we define an ensemble of LDPC codes with the variable and check degree vectors (sequences) $\underline{\lambda} = [\lambda_2, ..., \lambda_L]$ and $\underline{\rho} = [\rho_2, ..., \rho_R]$, where λ_i denotes the fraction of edges connected to variable nodes of degree i, and ρ_j denotes the fraction of edges connected to check nodes of degree j. Here L and R denote the maximum variable and the check node degree, respectively. Obviously, $\sum_{i=2}^{L} \lambda_i = \sum_{i=2}^{R} \rho_i = 1$. The code rate equals $r = 1 - \frac{\sum_{j=2}^{R} \rho_j / j}{\sum_{i=2}^{L} \lambda_i / i}$. Using density evolution [5][7], given the distribution of the degree sequences $\{\underline{\lambda}, \underline{\rho}\}$, we can evaluate the noise tolerance threshold σ^*, where σ^* is the maximal noise standard deviation for which the code will be decoded with zero probability of error.

We denote the average probability density function (pdf) of the log-likelihood-ratio message emitted from a check node to a variable node after ℓ iterations by $f_c^{(\ell)}$. The average pdf of the message emitted from a variable node to a check node is denoted by $f_v^{(\ell)}$. The average pdf obtained by evolving the density $f_v^{(\ell)}$ through the channel trellis is denoted by $f_o^{(\ell)}$. Denoting the convolution operator by $\otimes$ and the convolution of i–1 pdf's by $\otimes_{k=1}^{i-1}$, we have [7]

$$f_v^{(\ell)}(\xi) = \left[\sum_{i=2}^{L} \lambda_i \left(\otimes_{k=1}^{i-1}\left(f_c^{(\ell-1)}(\xi)\right)\right)\right] \otimes f_o^{(\ell-1)}(\xi) \ , \quad (3)$$

and
$$f_c^{(\ell)}(\xi) = \sum_{i=2}^{R} \rho_i \varepsilon_c^{i-1}\left(f_v^{(\ell)}(\xi)\right) \ . \quad (4)$$

Here $\varepsilon_c^{i-1}(f_v^{(\ell)}(\xi))$ is the symbolic notation for the average message density obtained by evolving the density $f_v^{(\ell)}(\xi)$ through a check node of degree i (see [7]). The average probability that a variable-to-check edge carries an erroneous message after ℓ iterations equals

$$\begin{aligned} p_\ell &= \int_{-\infty}^{0} f_v^{(\ell)}(\xi) d\xi \\ &= \int_{-\infty}^{0} \left(\sum_{i=2}^{L} \lambda_i \left(\otimes_{k=1}^{i-1}\left(f_c^{(\ell-1)}(\xi)\right)\right)\right) \otimes f_o^{(\ell-1)}(\xi) d\xi \\ &= \int_{-\infty}^{0} \sum_{i=2}^{L} \lambda_i f_{\ell,i}(\xi) d\xi \ . \end{aligned} \quad (5)$$

Here we use $f_{\ell,i}(\xi)$ as short notation for $\left(\otimes_{k=1}^{i-1}\left(f_c^{(\ell-1)}(\xi)\right)\right)$ $\otimes f_o^{(\ell-1)}(\xi)$.

III. Optimization

We slightly change the scenario of density evolution described in [7]. Instead of running the BCJR algorithm after each *variable-to-check density update*[1], we apply the BCJR once, then the variable-to-check density update $K \geq 1$ times, then the BCJR once, and so forth.

Following [5] we rewrite the probability of error (5) as

$$p_\ell = \sum_{i=2}^{L} A_{\ell,i} \lambda_i \ , \quad (6)$$

where $A_{\ell,i}$ represents the probability of error after ℓ iterations when $\underline{\lambda}$ is used for the first $\ell-1$ iterations and the singleton $\lambda_i = 1$ is used in the ℓ^{th} iteration, i.e.,

$$A_{\ell,i} = \int_{-\infty}^{0} f_{\ell,i}(\xi) d\xi \ . \quad (7)$$

When $\ell-1$ iterations are performed with parameters $\{\underline{\lambda}, \underline{\rho}\}$ and in the ℓ^{th} iteration $\underline{\lambda}$ is changed to $\underline{\lambda}'$, the probability of error equals

$$p'_\ell = \sum_{i=2}^{L} A_{\ell,i} \lambda'_i \ . \quad (8)$$

Multiplicative factors $A_{\ell,i}$ in equations (6) and (8) are identical because we do not update $f_o^{(\ell)}$ through the channel trellis after $\underline{\lambda}$ is changed to $\underline{\lambda}'$.

The density evolution algorithm with calculations of quantities relevant for the optimization is given by

Algorithm I (density evolution)

1. initialize $\ell = 1, f_c^{(0)} = \delta(\xi), f_o^{(0)} = \delta(\xi)$.
2. update $f_v^{(\ell)}$ according to (3).
3. update $f_c^{(\ell)}$ according to (4).
4. compute p_ℓ and $A_{\ell,i}$ according to (6) and (7).
5. if $l \bmod K = 0$ update $f_o^{(\ell)}$ through the channel trellis; otherwise $f_o^{(\ell)} = f_o^{(\ell-1)}$.
6. increment ℓ and repeat steps 2-6 until ℓ reaches a predefined number of iterations m.

[1] A variable-to-check density update is one iteration of density evolution through the LDPC portion of the graph.

Using **Algorithm I** we perform the optimization in a very similar fashion to [5]. That is, given an initial degree sequence $\underline{\lambda}$ we optimize the degree sequence $\underline{\lambda}'$ by minimizing the approximate number of iterations $G(\underline{\lambda}')$ needed to take the probability of error from p_0 down to p_m, where

$$G(\underline{\lambda}') = \sum_{\ell=1}^{m} \frac{p'_\ell - p_\ell}{p_{\ell-1} - p_\ell} \ .$$

Denoting an arbitrarily chosen small constant by δ we conduct the minimization under the following constraints [5][6]

$$\lambda'_i \geq 0, \ \ \sum_{i=2}^{L} \lambda'_i = 1, \ \ \sum_{i=2}^{L} \frac{\lambda'_i}{i} = \sum_{i=2}^{L} \frac{\lambda_i}{i}, \ \ p'_\ell \leq p_{\ell-1}, \ (1 \leq \ell \leq m)$$
$$|p_\ell - p'_\ell| \leq \delta \cdot \max(0, p_{\ell-1} - p_\ell), \quad (1 \leq \ell \leq m; \ 0 < \delta \ll 1). \quad (9)$$

This can be accomplished by means of linear programming.

The optimization of the check degree sequence $\underline{\rho}$ is analogous to the method for optimizing $\underline{\lambda}$ described above. Alternatively, we can limit the search space for $\underline{\rho}$ by allowing this degree sequence to have only a few non-zero entries. This often gives satisfactory results and is advantageous with respect to the algorithm execution time.

Although $p_{\ell-1} \geq p_\ell$ always holds for continuous message pdf's [5], this is not always the case when the pdf's are discretized. To minimize the number of values of ℓ for which the last constraint in (9) represents an equality, we need to initialize σ to a sufficiently small value. On the other hand, for very low values of σ, the values of p_ℓ, even for small ℓ's, become comparable to the computer precision floors. Therefore, we introduce a probability limit $p_{\lim}$ to ensure that σ is not too low. As we update (improve) $\underline{\lambda}$ we increase σ such that $p_{\ell-1} > p_\ell > 0$ for almost all values of $1 \leq \ell \leq m$. The degree sequence optimization algorithm is given by

Algorithm II (degree optimization)

1. initialize $\sigma = \sigma_0$, and $p_{\lim}$ (say, $p_{\lim} = 10^{-5}$).
2. initialize $\underline{\lambda}$ and $\underline{\rho}$, such that $r = 1 - \frac{\sum_{j=2}^{R} \rho_j / j}{\sum_{i=2}^{L} \lambda_i / i}$ equals the desired code rate.
3. compute p_l and $A_{l,i}$, $(1 \leq l \leq m, \ \ 2 \leq i \leq L)$ using density evolution (**Algorithm I**) with the degree sequences $\underline{\lambda}$ and $\underline{\rho}$, assuming the noise standard deviation equals σ.
4. find $\underline{\lambda}'$ that minimizes $G(\underline{\lambda}')$ under the given constraints (9) using a linear program and update $\underline{\lambda} \leftarrow \underline{\lambda}'$.
5. if $p_m \leq p_{\lim}$ update $\sigma \leftarrow \sigma + \delta\sigma$; goto step 3.

Algorithm II terminates when the updated degree sequences can not further increase σ. Since this is a local optimization method, we need to restart with different initial $\underline{\lambda}$ and $\underline{\rho}$ until satisfactory sequences are found.

IV. Results

We performed our optimization method on the Dicode ($h(D) = 1 - D$) and the EPR4 ($h(D) = (1-D)(1+D)^2$) channel. We empirically chose K to maximize the speed of the density evolution and the spreading of probabilities p_0 to p_m. A good choice is $K = 5$ for the Dicode channel, and $K = 3$ for the EPR4 channel (longer channel memory tends

Dicode channel $r=0.7$			EPR4 channel $r=0.7$		
i	λ_i	ρ_i	i	λ_i	ρ_i
2	0.2032	0.0004	2	0.2022	
3	0.2298	0.0002	3	0.2244	
5	0.1397		4	0.0269	
6	0.0077		6	0.0417	
15		0.6252	10		0.1934
16		0.3588	11	0.0048	0.1023
30		0.0154	12	0.0007	
47	0.1271		14		0.0658
48	0.2925		23		0.3138
			24		0.3247
			42	0.1576	
			43	0.3157	
			50	0.0260	
threshold $\sigma^*=0.6586$			threshold $\sigma^*=0.6269$		
distance to $C_{i.u.d}$ 0.14dB			distance to $C_{i.u.d}$ 0.15dB		

TABLE I

Good degree sequences for the Dicode and the EPR4 channel.

to imply shorter K). We stopped the optimization when we reached thresholds within 0.15dB of the i.u.d. capacity. Examples of optimized degree sequences for the design code rate $r=0.7$ and their respective thresholds (σ^*) are shown in Table I. More thresholds are plotted next to the i.u.d. capacities in Figure 1. All the results can be further improved, albeit by a slight margin, given enough time and patience. The maximum degrees were set to $L=50$ and $R=50$. Increasing L and R results in slight improvements, but the construction of codes with finite block lengths becomes more difficult.

We used the degree sequences from Table I to construct irregular LDPC coset codes of length 10^6 bits at the code rate $r=0.7$. Coset code simulation results are presented in Figure 2. We see that these codes achieve the bit error rate 10^{-6} (we simulated 1000 blocks) at SNR's less than 0.22dB away from the i.u.d. capacities.

Based on the experiments it is tempting to conclude (we cannot offer a proof) that the optimal LDPC codes asymptotically approach the i.u.d. capacities as the maximum degrees (L,R) and the block lengths grow infinitely large.

V. Conclusion

We have presented an algorithm that optimizes the degree sequences of the LDPC message passing decoder for partial response channels. Our strategy is based on adapting the optimization method for memoryless channels [5][6] to the density evolution algorithm for channels with ISI memory [7]. With this strategy we constructed irregular LDPC codes with the noise thresholds within 0.15dB of the i.u.d. capacity. Simulations show that coset codes of length 10^6 bits for both the Dicode and the EPR4 channel at code rate $r=0.7$ achieve bit error rates 10^{-6} at SNRs less than 0.22dB away from the i.u.d. capacities.

The optimization method we presented in this letter can also be used to optimize outer LDPC codes in a serially concatenated probabilistic coding scheme that yields codes that surpass the i.u.d. channel capacities [10].

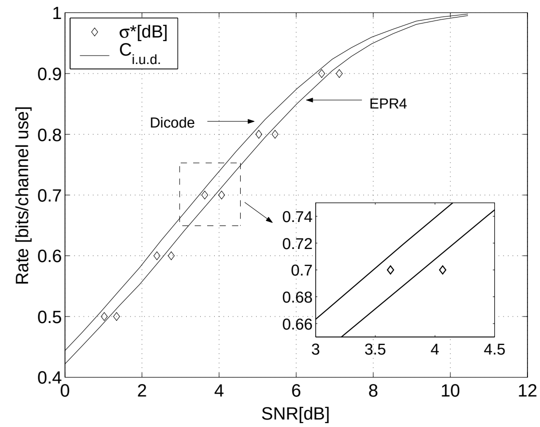

Fig. 1. The i.u.d. capacity curves and the LDPC code thresholds for the Dicode and the EPR4 channels. The thresholds are close to $C_{i.u.d.}$ over the entire range of code rates from $r=0.5$ to $r=0.9$

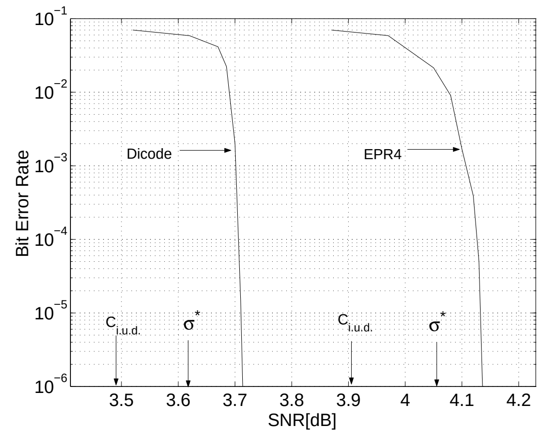

Fig. 2. Bit error rate vs. SNR for the Dicode and the EPR4 channels; code length 10^6 bits; code rate $r=0.7$

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in *Proc. IEEE Int. Conf. on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 1993.

[2] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low-density parity-check codes,” *Electronic Letters*, vol. 32, pp. 1645–1646, 1996.

[3] R. G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press, 1962.

[4] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Improved low density parity check codes using irregular graphs and belief propagation,” in *Proc. IEEE Int. Symp. Inform. Theory*, (Cambrdge, MA), p. 117, Aug. 1998.

[5] T. Richardson, A. Shokrollahi, and R. Urbanke, “Design of capacity-approaching low-density parity-check codes,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, February 2001.

[6] S.-Y. Chung, G. D. Forney, Jr., T. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” *IEEE Communications Letters*, vol. 47, pp. 58–60, February 2001.

[7] A. Kavčić, X. Ma, and M. Mitzenmacher, “Binary intersymbol interference channels: Gallager codes, density evolution and code performance bounds.” accepted for publication in *IEEE Trans. Inform. Theory*; available at http://hrl.harvard.edu/~kavcic/publications.html, Feb 2001.

[8] D. Arnold and H.-A. Loeliger, “On the information rate of binary-input channels with memory,” in *Proceedings IEEE ICC 2001*, (Helsinki, Finland), June 2001.

[9] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, “On the achievable information rates of finite state ISI channels,” in *Proceedings IEEE Global communications Conference 2001*, (San Antonio, Texas), November 2001.

[10] X. Ma, N. Varnica, and A. Kavčić, “Matched information rate codes for binary ISI channels.” *International Symposium on Information Theory*, Lausanne, Switzerland, July 2002.