Noise Thresholds for Discrete LDPC Decoding Mappings

Brian M. Kurkoski, Kazuhiko Yamaguchi, Kingo Kobayashi
Dept. of Information and Communication Engineering
University of Electro-Communications,
1-5-1 Chofugaoka, Chofu, Tokyo, Japan
{kurkoski,yama,kingo}@ice.uec.ac.jp

***Abstract*— For decoding low-density parity-check (LDPC) codes on discrete memoryless channels, a method to quantize messages and to find message-passing decoding functions for the variable and check nodes is developed. These are used to obtain noise thresholds by density evolution. The message-passing decoding alphabet is restricted to be discrete with a fixed maximum alphabet size. Discrete quantization is required to obtain this fixed alphabet size; a greedy algorithm which uses the mutual information between the code bit and message is presented. It is argued that using this message-passing decoding framework is more efficient for approaching channel capacity than simply quantizing the belief-propagation algorithm. This method is evaluated using regular LDPC codes on the binary symmetric channel. Using a maximum alphabet size of 16 (4 bits), noise thresholds close to those of belief propagation are obtained.**

I. Introduction

Low-density parity check codes allow for reliable communications arbitrarily close to the theoretical capacity of a number of channels models [1]. The decoding algorithm for LDPC codes iteratively passes messages between variable nodes and check nodes. At each node, outgoing messages are computed from incoming messages. Under belief-propagation (or sum-product) decoding, the messages are probabilities, and the node implements functions derived from properties of probability e.g. Bayes' rule. Belief propagation decoding is bit-error optimal on cycle-free graphs, and has excellent performance on graphs even with cycles [2] [3]. Belief-propagation decoding, or an algorithm derived from it, is widely used in practice.

However, in their landmark paper on LDPC codes, Richardson and Urbanke described a more general message-passing decoding of LDPC codes, where the messages are from an arbitrary alphabet, and the check and variable nodes have arbitrary decoding mappings from input alphabets to output alphabets [4]. Belief-propagation decoding is the most important example of message-passing decoding. Message-passing decoding is reviewed in Section II.

The subject of this paper is the development of discrete message-passing decoding mappings which are not derived from quantized belief-propagation, and the determination of the corresponding noise thresholds. A reasonable design criteria is to maximize the mutual information between a code bit and its message, since the goal is to communicate as close as possible to channel capacity, and channel capacity is a maximization of mutual information. Further, EXIT charts, an iterative decoding analysis tool, are also based upon mutual information [5]. A discrete memoryless channel is assumed, and the message alphabets are also discrete.

Before discussing density evolution, the quantization of the output of a discrete memoryless channel is considered. Specifically, an arbitrary discrete memoryless channel has binary inputs X and output Y. Then Y is quantized to another variable Z. The quantizer should be selected to maximize mutual information $I(X;Z)$, subject to the constraint that the alphabet size of Z is no larger than a fixed value, M. We propose a greedy combining quantization algorithm, where maximizing mutual information is used as the one-step combining criteria; it is not, however, guaranteed to be globally optimal. This quantization problem and the greedy combining approach are presented in Section III.

The rest of the paper is given to using this quantization approach to obtain message distributions and noise thresholds by density evolution. Each step of decoding is decomposed into a sequence of mappings in two discrete variables. Each node represents a function of code symbols, either equality at the variable node or addition at the check node. Given this function, it is straightforward to compute the output message distribution. At each stage of the decoding algorithm, a new joint distribution is formed. If iterations are allowed to progress, this quantized distribution will strongly resemble the one produced by standard density evolution. However, with each decoding step, the alphabet size will increase, making analysis difficult. Thus, a large discrete alphabet is mapped onto a smaller alphabet, which is the quantization of a discrete memoryless channel. With a reduced alphabet size, density evolution proceeds, with quantization performed at each step. Whereas traditional density evolution requires the channel satisfy a symmetry condition, in this paper, joint densities are used and the approach is valid for arbitrary discrete memoryless channels. This is described in detail in Section IV.

Richardson and Urbanke also described a specific decoder map for message-passing decoding which uses an alphabet of size three and has intuitive decoding mappings. This

This work was supported in part by the Ministry of Education, Science, Sports and Culture; Grant-in-Aid for Scientific Research (C) number 19560371.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

Algorithm E had modest performance loss relative to the more complicated belief propagation on the binary symmetric channel [4]. The proposed techniques are more general than Algorithm E in that the alphabet size M is arbitrary. However, the proposed algorithm with $M = 3$ has the same noise thresholds as Algorithm E. By increasing the alphabet size, the noise thresholds of belief-propagation decoding can be more closely approached.

In prior work by Lee and Thorpe, an LDPC decoder using a quantizer which maximizes mutual information is also designed. The decoder produced impressive results, coming within 0.1 dB of belief propagation performance using 4 bits per message, when using a block-length 8000 LDPC code on the binary-input AWGN channel. However, the technique for selecting the quantizer was not described, and they stated that a significant amount of hand optimization was required [6].

The density evolution algorithm in this paper is explicitly described and requires only one parameter: the maximum message alphabet size, M. Our attention is restricted to finding thresholds for discrete memoryless channels. It is found that for the binary symmetric channel, using about 16 message levels (corresponding to 4 bits), that noise thresholds using the techniques proposed in this paper can come very close to those obtained by belief-propagation decoding. Numerical noise thresholds for regular LDPC codes of various rates on the binary symmetric channel are given in Section V, as well as a discussion contrasting the proposed approach with quantized belief-propagation decoding.

II. LDPC Codes

The following notation is used. A discrete random variable X takes on a value x from an alphabet $\mathcal{X}$, with probability $Pr(X = x)$ or $P_X(x)$. The alphabet size is $|\mathcal{X}|$.

A. LDPC Codes

A binary LDPC code is the set of binary $\{0, 1\}$ code vectors $(X_1, \ldots, X_n)$ of n elements such that over the binary field,

$$(X_1, \ldots, X_n)H^t \quad = \quad 0, \tag{1}$$

where H is a sparse parity check matrix with constant column weight d_{v} and row weight d_{c}. The rate of the code is $1-d_{\mathsf{v}}/d_{\mathsf{c}}$. From here, infinite-length block codes are assumed so that the effects of loops in the bipartite graph for LDPC codes can be ignored, and density evolution is exact.

The binary code symbol X is transmitted over a two-input discrete memoryless channel, received as W from an output alphabet $\mathcal{W}$. The channel transition probability is $Pr(W = w|X = x)$.

B. Message-Passing Decoding Algorithm

In message-passing decoding [4], the variable-to-check messages R are from the message alphabet $\mathcal{R}$. Similarly, the check-to-variable messages L are from the message alphabet $\mathcal{L}$.

At iteration ℓ, the check node with degree d_{c} computes outgoing message $L_{d_{\mathsf{c}}}$ using the incoming messages $R_1, \ldots, R_{d_{\mathsf{c}}-1}$, by a mapping function $\chi^{(\ell)} : \mathcal{R}^{d_{\mathsf{c}}-1} \to \mathcal{R}$. A mapping finds the message for a variable from the messages for two or more other variables.

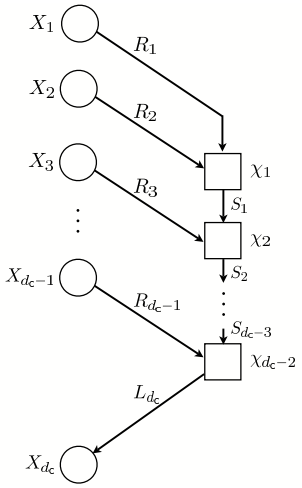


Fig. 1. Check node: mappings χ_i, input messages R_i, internal messages S_i, output message $L_{d_{\mathsf{c}}}$.

As a simplification, the mapping χ is decomposed into a series of mappings, $\chi_1, \ldots, \chi_{d_{\mathsf{c}}-2}$, each in two variables. Introduce $S_i, i = 1, \ldots, d_{\mathsf{c}} - 3$ which are messages for the partial sum $X_1 + \cdots + X_{i+1}$ (at the check node, $X_1 + \cdots + X_{d_{\mathsf{c}}} = 0$). Thus, the decomposition is:

$$\chi_1^{(\ell)} \quad : \quad \mathcal{R} \times \mathcal{R} \to \mathcal{S}_1, \tag{2}$$

for $i = 2, 3, \ldots, d_{\mathsf{c}} - 3$,

$$\chi_i^{(\ell)} \quad : \quad \mathcal{S}_{i-1} \times \mathcal{R} \to \mathcal{S}_i, \tag{3}$$

and finally,

$$\chi_{d_{\mathsf{c}}-2}^{(\ell)} \quad : \quad \mathcal{S}_{d_{\mathsf{c}}-3} \times \mathcal{R} \to \mathcal{L}. \tag{4}$$

The check node mapping is shown in Fig 1.

Similarly, at iteration ℓ, the variable node with degree d_{v} finds the outgoing message $R_{d_{\mathsf{v}}}$ using the channel value W and incoming messages $L_2, \ldots, L_{d_{\mathsf{v}}-1}$, by a mapping function $\Phi^{(\ell)} : \mathcal{W} \times \mathcal{L}^{d_{\mathsf{v}}-1} \to \mathcal{R}$.

As a simplification, the mapping $\Phi^{(\ell)}$ is decomposed into a series of mappings, $\Phi_1, \ldots, \Phi_{d_{\mathsf{v}}-1}$, each in two variables. Intermediate variables A_i are introduced as:

$$\Phi_1^{(\ell)} \quad : \quad \mathcal{W} \times \mathcal{L} \to \mathcal{A}_1, \tag{5}$$

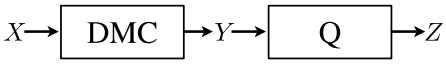
for $i = 2, 3, \ldots, d_{\mathsf{v}} - 2$

$$\Phi_i^{(\ell)} \quad : \quad \mathcal{A}_{i-1} \times \mathcal{L} \to \mathcal{A}_i, \tag{6}$$

$$\Phi_{d_{\mathsf{v}}-1}^{(\ell)} \quad : \quad \mathcal{A}_{d_{\mathsf{v}}-2} \times \mathcal{L} \to \mathcal{R}. \tag{7}$$

In this paper, the channel output alphabet $\mathcal{W}$ and the message-passing alphabets $\mathcal{R}$, $\mathcal{L}$, $\mathcal{S}$ and $\mathcal{A}$ are discrete. However, belief-propagation decoding can be described using the above framework. The message alphabets are probabilities, that is $\mathcal{R} = \mathcal{L} = [0, 1]$. The check node decoding map χ can be obtained by sum-product rules. The variable node decoding

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.



Find:

$$Q \quad = \quad \arg\max_{Q} I(X;Z),$$

such that,

$$|\mathcal{Z}| \quad \leq \quad M$$

Fig. 2. Quantizer for the output of a DMC.

map Φ is obtained by "belief propagation" i.e. multiplication of probabilities.

III. Quantization of the Output of a DMC

A. Preliminaries

In this section, we consider the following general problem, which will be of interest in finding probability densities for density evolution.

A symbol X distributed as $Pr(X = x)$ is transmitted over a discrete memoryless channel with $Pr(Y = y|X = x)$. The channel output Y is then quantized to Z by a quantizer Q, as shown in Fig. 2. This problem has been formulated as communication over a discrete memoryless channel because it is intuitively appealing, however note that $Pr(X = x)$ is given; this is distinct from channel capacity computation where the objective often is to find $Pr(X = x)$.

The problem is to find the quantization mapping Q,

$$Q \quad : \quad \mathcal{Y} \to \mathcal{Z} \tag{8}$$

which maximizes the mutual information between X and Z,

$$Q = \arg\max_{Q} I(X;Z) \text{ subject to } |\mathcal{Z}| \leq M, \tag{9}$$

where M is the fixed maximum alphabet size of Z.

The statement of this problem appears similar to the computation of minimum or maximum mutual information as proposed by Arimoto and Blahut, but is distinct. The computation of rate-distortion functions in source coding, and channel capacity in channel coding, can be found by an alternating minimization or maximization algorithm [7, Sec. 13.7]. However, we have not found a formulation of (9) which can be put into the alternating maximization form.

As an aside, note that it is possible to write the quantizer Q as a matrix $Q_Y = q_{ij}$ where $q_{ij} = 1$ if $i = (y_1, y_2)$ quantizes to $j = z$, and $q_{ij} = 0$ otherwise. If the joint probability distribution P_{XY} is written as an $|\mathcal{X}|$-by-$|\mathcal{Y}|$ matrix, then the joint distribution P_{XZ} is the $|\mathcal{X}|$-by-$|\mathcal{Z}|$ matrix:

$$P_{XZ} \quad = \quad P_{XY} \cdot Q_Y. \tag{10}$$

It is this joint distribution P_{XZ} which will be used in density evolution.

B. Quantization by Greedy Combining

A greedy combining algorithm is proposed to find a quantizer Q. Each step of the algorithm constructs temporary random variable T which is an approximation of Y. At each iteration, the joint distribution of T and X is replaced with by a new joint distribution which has one fewer values over the alphabet $\mathcal{T}$. This is repeated iteratively until the alphabet size of T is reduced to M.

A quantizer Q may be found by greedy combining:

1) Initialize. $\mathcal{T} \leftarrow \mathcal{Y}$ and $Pr(X = x, T = i) \leftarrow Pr(X = x, Y = i)$ for all x, i.
2) For each distinct pair $t, s \in \mathcal{T}$, construct a temporary random variable (or message) $T_{t,s}$, with $|\mathcal{T}_{t,s}| = |\mathcal{T}| - 1$, obtained by combining t, s to a single element $z \in \mathcal{T}_{t,s}$. There are $\binom{|T|}{2}$ such pairs. The remaining elements are not modified. The joint distribution is:

$$Pr(X = x, T_{t,s} = i) \tag{11}$$
$$= \begin{cases} Pr(X = x, T = i), \text{ if } i \neq z \\ Pr(X = x, T = t) \\ \qquad + Pr(X = x, T = s), \text{ if } i = z. \end{cases}$$

The mutual information for the pairing of t, s is:

$$I_{t,s} \quad = \quad I(X; T_{t,s}). \tag{12}$$

3) Select the pair $\widehat{t}, \widehat{s}$, which maximizes the mutual information:

$$\widehat{t}, \widehat{s} \quad = \quad \arg\max_{t,s} I_{t,s} \tag{13}$$

4) Accept this combination of $\widehat{t}$ and $\widehat{s}$ as the new approximation, $\mathcal{T} \leftarrow \mathcal{T}_{\widehat{t},\widehat{s}}$ and $Pr(X = x, T = i) \leftarrow Pr(X = x, T_{\widehat{t},\widehat{s}} = i)$ for all x, i.
5) If $|\mathcal{T}| > M$, go to Step 2. Otherwise accept T as the final approximation, that is $\mathcal{Z} \leftarrow \mathcal{T}$ and $Pr(X = x, Z = i) \leftarrow Pr(X = x, T = i)$ for all x, i.

This is a greedy search algorithm, and thus no claims about its optimality in maximizing mutual information can be made. Nonetheless, as will be shown in the numerical results section, the results obtained using this algorithm are reasonable.

The complexity of this algorithm is proportional to $|\mathcal{Y}|^2$, due to the initial computation of the mutual information between all distinct pairs in $\mathcal{Y}$. Following steps only recompute the mutual information between the new element all the unchanged elements.

IV. Message-Passing Density Evolution for Binary LDPC Codes

In the density evolution method presented here, joint distributions are tracked rather than conditional message distributions. Subsection IV-A describes how the joint distributions are found at any given decoding step. Subsection IV-B describes how these are combined to perform density evolution.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

A. Finding Joint Distributions

In generic terms, each mapping $\mathcal{Y}_1 \times \mathcal{Y}_2 \to \mathcal{Z}$ given in (2) to (7) describes decoding for a deterministic function $X = f(X_1, X_2)$. The joint distributions $Pr(X_1, Y_1)$ and $Pr(X_2, Y_2)$ are given. This subsection describes how to obtain the joint distribution $Pr(X, Y)$ where $Y = (Y_1, Y_2)$. Then, given $Pr(X, Y)$, the quantizer $Q : \mathcal{Y} \to \mathcal{Z}$ is found, to obtain the joint distribution $Pr(X, Z)$.

At the check node, consider a variable X which is a deterministic function of variables X_1 and X_2, specified by $X = f(X_1, X_2) = X_1 + X_2$. At an arbitrary point in message computation, the messages for X_1 and X_2 are Y_1 and Y_2, respectively, and the node function finds Y, the message for X. Let the alphabet of Y be $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2$. The joint distributions $Pr(X_1, Y_1)$ and $Pr(X_2, Y_2)$ are independent,

$$Pr(X_1, X_2, Y_1, Y_2) \quad = \quad Pr(X_1, Y_1) \cdot Pr(X_2, Y_2),$$

as is usually assumed in density evolution. The joint distribution $Pr(X, Y)$ is found as follows:

$$Pr\big(X = x, Y = (y_1, y_2)\big) = \sum_{x_1, x_2 : f(x_1, x_2) = x} Pr(X_1 = x_1, Y_1 = y_1) \cdot Pr(X_2 = x_2, Y_2 = y_2). \tag{14}$$

For the variable node, the above procedure is largely the same. The code symbol X is transmitted over two independent discrete memoryless channels and the received symbols are Y_1 and Y_2, with joint distributions $Pr(X, Y_1)$ and $Pr(X, Y_2)$. Again, letting $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2$, the joint distribution is:

$$\begin{aligned} Pr(X, Y) &= Pr(X, Y_1, Y_2) \\ &= \frac{1}{Pr(X)} Pr(X, Y_1) \cdot Pr(X, Y_2). \end{aligned}$$

The term $Pr(X_1)$ can be ignored because of normalization. Eqn. (14) may be applied to the variable node with the following deterministic function f,

$$f(x_1, x_2) \quad = \quad \frac{1}{2}\big(x_1 + x_2\big), \tag{15}$$

where addition is over the reals, rather than the finite field. The above function has the property that it is 0 (respectively 1) only when x_1 and x_2 are both 0 (respectively 1).

The above may be regarded as transmission over two discrete memoryless channels, where X_1 and X_2 are received as Y_1 and Y_2. In the above procedure, we first found $Pr(X_1, X_2, Y_1, Y_2)$, and then applied $f(\cdot)$ to get the joint distribution $Pr(X, Y)$. This is shown in Fig. 3-(a) for check nodes and (b) for variable nodes.

Note that the alphabet size $|\mathcal{Y}|$ is equal to $|\mathcal{Y}_1| \cdot |\mathcal{Y}_2|$. If this is applied recursively, as will be done in density evolution, then the alphabet size will grow rapidly. Thus, quantization is introduced. Specifically, the greedy quantization algorithm is applied to this joint distribution, which produces a discrete random variable Z and its joint distribution $Pr(X, Z)$ which is will be used in the following step of density evolution.

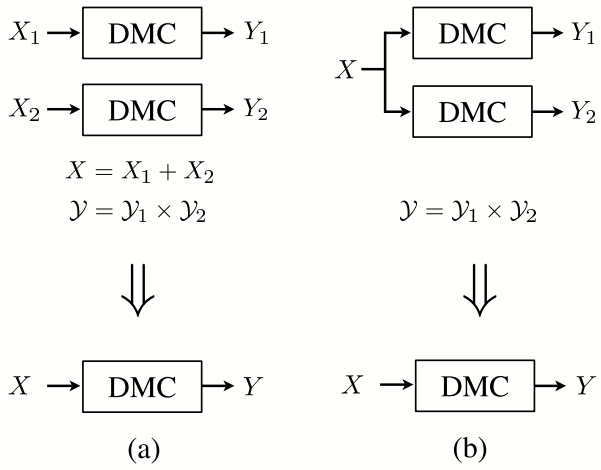


Fig. 3. Constructing the composition of the output of two DMCs. (a) Check node (b) variable node.

B. Density Evolution

Density evolution finds the joint densities for the messages R, L, S and A that were described in Section II. Those joint densities are $Pr(X, R)$, $Pr(X, L)$, $Pr(X, S)$ and $Pr(X, A)$, respectively. The only parameter for this density evolution is M, which is the maximum alphabet size allowed after quantization has been performed.

For each iteration $\ell = 1, 2, \cdots$, iterate between the check node and variable node as follows.

Check Node The input to the check node step is the joint distribution $Pr(X, R)$. On the first iteration only, the channel joint distribution $Pr(X, W)$ is used in place of the variable-to-check distribution $Pr(X, R)$. In particular, $\mathcal{R} = \mathcal{W}$, and $Pr(X = x, R = r) = Pr(X = x, W = r)$, for all $x \in \mathcal{X}, r \in \mathcal{R}$.

1) Using $Pr(X, R)$ twice, compute $Pr(X, S_1)$.
2) For $i = 2, \ldots d_{\mathsf{c}} - 3$, use $Pr(X, S_{i-1})$ and $Pr(X, R)$ to compute $Pr(X, S_i)$.
3) Use $Pr(X, S_{d_{\mathsf{c}}-3})$ and $Pr(X, R)$ to compute $Pr(X, L)$.

The output is the joint distribution $Pr(X, L)$.

Variable Node The inputs to the variable node step are the message distribution $Pr(X, L)$ and the channel distribution $Pr(X, W)$.

1) Use $Pr(X, W)$ and $Pr(X, L)$ to compute $Pr(X, A_1)$.
2) For $i = 2, \ldots d_{\mathsf{v}} - 2$, use[1] $Pr(X, A_{i-1})$ and $Pr(X, L)$ to compute $Pr(X, A_i)$.
3) Use $Pr(X, A_{d_{\mathsf{v}}-2})$ and $Pr(X, L)$ to compute $Pr(X, R)$.

The output is the joint distribution $Pr(X, R)$.

Termination The above process iterates. Convergence is declared if the mutual information $I(X; R)$ approaches 1. If instead a maximum number of iterations is reached, then non-convergence is declared.

[1]For Algorithm E, use $M = 6$ in this step only; in all other steps, $M = 3$.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

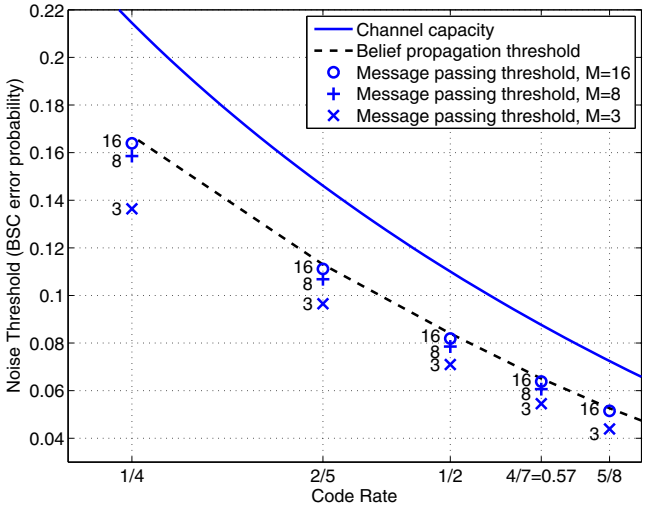


Fig. 4. Noise thresholds on the binary symmetric channel for the proposed message-passing density evolution technique, with maximum alphabet size M ($M = 16$ corresponds to 4-bit messages). Standard belief propagation noise thresholds and channel capacity are also shown. Regular LDPC codes with variable node degree 3 are used.

Noise threshold The noise threshold for a single-parameter discrete memoryless channel is the value for the most degraded channel for which convergence is obtained.

In traditional belief-propagation density evolution, the marginal message probability distributions are used under the assumption that the all-zeros codeword is transmitted, rather than the joint distributions used here. The channel must satisfy a symmetry property for belief-propagation density evolution analysis to be valid for an arbitrary codeword.

In the joint density evolution presented here, no assumptions were made about the discrete memoryless channel, in particular channel symmetry was not required. However, we have not performed convergence analysis. Nonetheless, density evolution closely tracks the performance of large block length codes, and it is reasonable that this analysis holds for general channels. Also, while joint distributions, for example $Pr(X, R)$, have been used throughout, it is possible to use conditional distributions, for example $Pr(R|X)$, as well.

V. Numerical Results, Discussion and Conclusion

The proposed density evolution technique was applied to regular LDPC codes used on the binary symmetric channel, and the noise thresholds obtained are shown in Fig. 4. Belief propagation noise thresholds and channel capacity are also shown. Increasing the maximum alphabet size M improves the noise threshold. In particular, using $M = 16$ gives noise thresholds which are close to those of belief-propagation decoding. For $M = 3$, we obtain similar thresholds for Richardson and Urbanke's Algorithm E.

A byproduct of the proposed meothod is that the message-passing decoding maps $\chi_i^{(\ell)}$ and $\Phi_i^{(\ell)}$ can be created at each step. These are precisely the mapping functions required for decoding. Such maps would be of interest in finite-length LDPC decoding. If the asymptotic results shown in Fig. 4 extend to finite-length decoding, then it would be expected that using approximately 16 levels (four or fewer bits) to represent messages would provide minimal performance loss, consistent with the results of Lee and Thorpe [6].

Quantized belief-propagation decoders are of great practical importance and these can be viewed in the same framework of discrete message-passing decoding. A message-passing decoding map could be derived by first selecting quantization points for continuous-valued messages (probabilities or log-likelihood ratios), and then finding the decoding maps χ and Φ by using nearest-neighbor quantization. Such an approach generally requires an a priori choice of a quantizer, often selected uniformly for convenience. These schemes are not optimized for the channel distribution (although they are general enough for arbitrary channels), and disregard mutual information measures. On the other hand, for a fixed channel, we argue that the proposed density evolution analysis of message-passing decoding will be more efficient than a quantized belief-propagation decoder. The quantization points are selected non-uniformly. The quantization maps are designed using a mutual information criteria, which is appropriate for channel coding.

In conclusion, a density evolution technique to compute noise thresholds using not belief-propagation decoding, but instead more general message-passing decoding has been proposed. Message-passing decoding maps were found by a quantization algorithm which uses mutual information criteria; a greedy optimization algorithm to do this was proposed. For the binary symmetric channel, it was found that using a maximum message alphabet size of $M = 16$ (4 bits) gave thresholds that were quite close to those obtained by standard belief-propagation density evolution.

References

[1] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 619–637, February 2001.

[2] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, February 2001.

[3] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, pp. 2282–2312, July 2005.

[4] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, February 2001.

[5] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, October 2001.

[6] J.-S. Lee and J. Thorpe, "Memory-efficient decoding of LDPC codes," *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pp. 459–463, 4-9 Sept. 2005.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 1991.