ML-Performance of Low-Density Parity-Check Codes

Heidi Steendam and Marc Moeneclaey

DIGCOM research group, TELIN Dept., Ghent University
Sint-Pietersnieuwstraat 41, 9000 GENT, BELGIUM
E-mail : {hs,mm}@telin.rug.ac.be

Abstract. In this paper, we derive the maximum likelihood (ML) performance for low-density parity check (LDPC) codes, considering BPSK and QPSK transmission over a Gaussian channel. We compare the theoretical ML performance with the performance of the iterative decoding algorithm. It turns out that the performance of the iterative decoding algorithm is close to the ML performance when the girth of the code is sufficiently high. When the girth of the code is equal to or smaller than 6, the decoding algorithm performs sub-optimal: the bit error rate (BER) obtained with the iterative decoding algorithm is much higher than the optimal BER that can be obtained when using ML decoding.

1 Introduction

Low-density parity check (LDPC) codes were introduced by Gallager [1] as a class of linear error-correcting block codes of which the check matrix is sparse. The original regular Gallager codes have a regularity constraint on the weight of the columns and the rows of the check matrix: the columns and the rows each contain a small fixed number of ones, j and k respectively. Gallager showed in [1] that random regular LDPC codes are asymptotically good and perform close to the Shannon capacity limit when the block length increases. However, LDPC codes were forgotten, as no practical decoding technique was available that was able to achieve the expected near-Shannon performance. Only recently, with the introduction of turbo codes [2], LDPC codes were rediscovered [3]. Similarly as for turbo codes [2], a decoding algorithm that was based on the belief propagation method [4] was proposed [3]. In this decoding algorithm, the aim is to compute the marginal posterior probability that a received bit is erroneous, given the information of the check matrix and the syndrome. This computation of the marginal posterior probability is done in an iterative way. When decoding the LDPC codes with this relatively simple and practical iterative decoding algorithm, it is shown in [5]-[7] that their empirical performance can approach the Shannon limit.

The search for a practical suboptimal decoding algorithm was necessary as the optimal (maximum likelihood, ML) decoder was too complex: the complexity of the ML decoder increases exponentially with the length of the information word. However, as the ML performance limits the performance of any decoding algorithm, it can be useful to compare the performance of the practical iterative decoding algorithm with the optimal ML performance. From the gap between the two, we can learn how to improve the code by carefully selecting the code parameters, such that the suboptimal decoding algorithm performs nearly optimal.

In this paper, we derive the maximum likelihood performance for LDPC codes, considering BPSK and QPSK transmission over a Gaussian channel, and compare the theoretical ML performance with the performance of the iterative decoding algorithm.

2 Low-Density Parity-Check Codes

A linear (K,N) binary block code is characterized by two parameters: the number K of information bits in a codeword and the length N of the codeword. The linear transformation that converts the information word of length K into the codeword is characterized by the N-by-K generator matrix G of the code. Defining $\mathbf{b} = (b_1 \dots b_K)$ as the information word and $\mathbf{c} = (c_1 \dots c_N)$ as the code word, the relationship between $\mathbf{b}$ and $\mathbf{c}$ is given by

$$\mathbf{c} = \mathbf{bG} \tag{1}$$

In this paper, we restrict our attention to systematic codes, i.e. the generator matrix is given by $\mathbf{G} = (\mathbf{P}|\mathbf{I_K})$, where $\mathbf{I_K}$ is a K-by-K identity matrix and $\mathbf{P}$ is the $N-K$-by-N check matrix. Hence, in this case, the last K bits of the code word are the information words. The code word $\mathbf{c}$ is transmitted over the Gaussian channel. At the receiver, the received word can be written as

$$\mathbf{r} = \mathbf{c} + \mathbf{e} \tag{2}$$

where $\mathbf{e}$ is the error word. The error word contains ones on positions where a transmission error occurred. To know if the transmitted word was received correctly, the syndrome $\mathbf{s} = (s_1 \dots s_M)$ is computed, i.e.

$$\mathbf{s} = \mathbf{rH}^T = \mathbf{cGH}^T + \mathbf{eH}^T = \mathbf{eH}^T \tag{3}$$

where $\mathbf{H}$ is the M-by-N check matrix and $\mathbf{GH}^T = \mathbf{0}$. If the received word is a codeword, the syndrome is all zero, i.e. $s_m = 0$, $m = 1, \dots, M$.

In a low-density parity-check code, the check matrix is sparse, i.e. it has a low density of ones. In this paper, we consider the regular LDPC codes introduced by Gallager [1]. In this type of codes, there is a regularity constraint on the weight of the columns and the rows of the check matrix: the columns and the rows each contain a small fixed number of ones, j and k respectively. An LDPC code can be represented by a bipartite graph. This bipartite graph consists of two types of nodes, i.e. the symbol nodes, which correspond to the bits of the received word, and the check nodes, which correspond to the bits of the syndrome. A connection (edge) between the symbol node r_n and the check node s_m indicates that the $(m,n)^{th}$ element of the check matrix equals 1. Hence, because of the regularity constraint, each symbol node (check node) is connected to exactly j check nodes (k symbol nodes), as indicated in figure 1. A cycle in the graph is a sequence of connections that starts and ends in the same node. In figure 2, a cycle of length 6 is shown. The minimum cycle length of the code is called the girth. It will be shown in section 4 that the girth of the code has an influence on the optimality of the iterative decoding algorithm.

Gallager presented a simple method to construct check matrices that satisfy the regularity constraints. In this method, the check matrix is characterized by three parameters, i.e. k which is the weight of the rows, j which is the weight of the columns, and p which is a prime number. The check matrix is constructed in the following way:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I_p} & \mathbf{I_p} & \cdots & \mathbf{I_p} \\ \mathbf{I_p} & \boldsymbol{\alpha} & \cdots & \boldsymbol{\alpha}^{k-1} \\ \vdots & \vdots & & \vdots \\ \mathbf{I_p} & \boldsymbol{\alpha}^{j-1} & \cdots & \boldsymbol{\alpha}^{(k-1)(j-1)} \end{bmatrix} \tag{4}$$

where $\mathbf{I_p}$ is the p-by-p identity matrix and $\boldsymbol{\alpha}$ is a p-by-p matrix that represents a single left or right cyclic shift, i.e.

$$\boldsymbol{\alpha} = \begin{bmatrix} 0&0&0&0&1 \\ 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \end{bmatrix} \qquad \boldsymbol{\alpha} = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 1&0&0&0&0 \end{bmatrix} \tag{5}$$

left cyclic shift   right cyclic shift

The matrix $\boldsymbol{\alpha}^x$ is the x^{th} power of the matrix $\boldsymbol{\alpha}$. The resulting check matrix is a pj-by-pk matrix. Hence, the length of the code word and the length of the syndrome are given by

$$\begin{aligned} N &= pk \\ M &= pj \end{aligned} \tag{6}$$

It turns out that in the check matrix (4), $j-1$ rows are linearly dependent on the other rows, such that the rank of the matrix equals $N-K = pj-(j-1)$. Hence, the length K of the information word equals

$$K = p(k-j)+j-1 \tag{7}$$

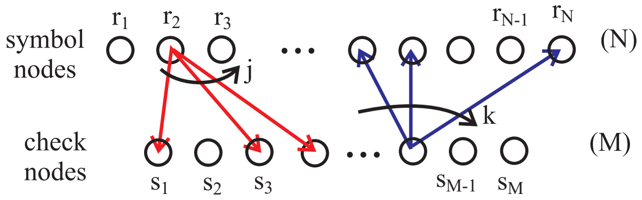

Fig. 1. Bipartite graph

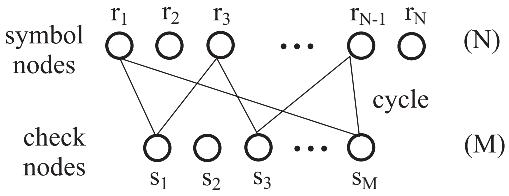

Fig. 2. Cycle of length 6

3 Maximum Likelihood Performance

The transceiver for BPSK (QPSK) transmission over a Gaussian channel is shown in figure 3. The code bits $\{c_n | n = 1, \ldots, N\}$ are first mapped on the symbols $\{t_n\}$ ($t_n \in \{-x_0, x_0\}$ for BPSK and $t_n \in \{x_0(\pm 1 \pm j)\}$ for QPSK). The sequence $\mathbf{t}$ is then transmitted over the Gaussian channel. The channel adds white Gaussian noise $\mathbf{w}$, with uncorrelated real and imaginary parts, each having a variance σ^2, resulting in the sequence $\mathbf{y} = \mathbf{t} + \mathbf{w}$. Based on the received sequence $\mathbf{y}$, a decision is taken about the transmitted code word. The ML decision rule is given by

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} d^2(\mathbf{y}, \mathbf{t}(\mathbf{c})) \tag{8}$$

where $d^2(\mathbf{a}, \mathbf{b})$ is the Euclidean distance between the sequences $\mathbf{a}$ and $\mathbf{b}$, and $\mathbf{t}(\mathbf{c})$ is the sequence of transmitted data symbols that correspond to the code word $\mathbf{c}$. Hence, the receiver selects the code word that corresponds to the sequence of symbols that is at minimum Euclidean distance of the received sequence $\mathbf{y}$. The bit error rate (BER) is given by

$$BER = \sum_{i,j=1;j\neq i}^{2^K} Pr(\hat{\mathbf{c}}_j | \mathbf{c}_i) Pr(\mathbf{c}_i) \frac{d_H(\mathbf{b}_j, \mathbf{b}_i)}{K} \tag{9}$$

where $Pr(\hat{\mathbf{c}}_j | \mathbf{c}_i)$ is the probability that the code word $\mathbf{c}_j$ is selected at the receiver when the code word $\mathbf{c}_i$ is transmitted, $Pr(\mathbf{c}_i)$ is the prior probability that the code word $\mathbf{c}_i$ is transmitted, $d_H(\mathbf{b}_j, \mathbf{b}_i$ is the Hamming distance between the information words $\mathbf{b}_j$ and $\mathbf{b}_i$), that correspond to the code words $\mathbf{c}_j$ and $\mathbf{c}_i$, respectively, and K is the length of the information word. In the following we

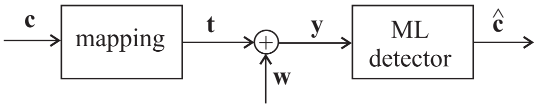

Fig. 3. BPSK/QPSK transmission over a Gaussian channel

assume that all information words, hence all code words, are equiprobable, i.e. $Pr(\mathbf{c}_i) = 1/2^K$. Considering the decision rule (8), the probability $Pr(\hat{\mathbf{c}}_j|\mathbf{c}_i)$ in (9) can be written as

$$Pr(\hat{\mathbf{c}}_j|\mathbf{c}_i) = Pr(d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_j)) < d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_\ell)); \ell = 1, \ldots, 2^K; \ell \neq j|\mathbf{c}_i) \tag{10}$$

Using the union bound approximation, a simple upper bound on the probability (10) can be found, i.e.

$$Pr(\hat{\mathbf{c}}_j|\mathbf{c}_i) \leq Pr(d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_j)) < d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_i))|\mathbf{c}_i) \tag{11}$$

Considering the received sequence $\mathbf{y}$ is a Gaussian variable, the probability $Pr(d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_j)) < d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_i))|\mathbf{c}_i)$ can be written as

$$Pr(d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_j)) < d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_i))|\mathbf{c}_i) = Q\left(\sqrt{\frac{d^2(\mathbf{t}(\mathbf{c}_j), \mathbf{t}(\mathbf{c}_i))}{2\sigma^2}}\right) \tag{12}$$

where

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \tag{13}$$

is the complementary error function. Hence, the BER is bounded by

$$BER \leq \frac{1}{2^K} \sum_{i,j=1;j\neq i}^{2^K} Q\left(\sqrt{\frac{d^2(\mathbf{t}(\mathbf{c}_j), \mathbf{t}(\mathbf{c}_i))}{2\sigma^2}}\right) \frac{d_H(\mathbf{b}_j, \mathbf{b}_i)}{K} \tag{14}$$

In the case of BPSK (QPSK) transmission, there exists a simple relationship between the Euclidean distance $d^2(\mathbf{t}(\mathbf{c}_j), \mathbf{t}(\mathbf{c}_i))$ between the BPSK (QPSK) symbols and the Hamming distance $d_H(\mathbf{c}_j, \mathbf{c}_i)$ between the corresponding code words, i.e.

$$d^2(\mathbf{t}(\mathbf{c}_j), \mathbf{t}(\mathbf{c}_i)) = (2x_0)^2 d_H(\mathbf{c}_j, \mathbf{c}_i) \tag{15}$$

Further, considering the energy per transmitted symbol equals $E_b = x_0^2$, and the noise power level $N_0/2 = \sigma^2$, (12) can be written as

$$Pr(d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_j)) < d^2(\mathbf{y}, \mathbf{t}(\mathbf{c}_i))|\mathbf{c}_i) = Q\left(\sqrt{2\frac{E_b}{N_0} d_H(\mathbf{c}_j, \mathbf{c}_i)}\right) \tag{16}$$

Considering (15) and (16), and taking into account the linearity of the code, the BER reduces to

$$BER \leq \sum_{j=2}^{2^K} Q\left(\sqrt{2\frac{E_b}{N_0} d_H(\mathbf{c}_j, \mathbf{0})}\right) \frac{d_H(\mathbf{b}_j, \mathbf{0})}{K} \tag{17}$$

where we assumed without loss of generality that $\mathbf{c}_1 = (0 \ldots 0)$ is the code word containing all zeroes, which corresponds to the information word $\mathbf{b}_1 = (0 \ldots 0)$. This upper bound on the ML performance appears a very tight upper bound on the actual ML performance for sufficiently high E_b/N_0 (typically when the BER is smaller than 10^{-3}, the upper bound is very close to the actual performance).

For high E_b/N_0, the sum in (17) is dominated by the term for which $d_H(\mathbf{c}_j, \mathbf{0})$ $= d_{H,min}$, where $d_{H,min}$ is the minimum Hamming distance of the code. Hence, for large E_b/N_0, the BER (17) can be approximated by

$$BER \approx A_{BER} Q\left(\sqrt{2\frac{E_b}{N_0} d_{H,min}}\right) \tag{18}$$

where

$$A_{BER} = \sum_{j=2}^{2^K} \frac{d_H(\mathbf{b}_j, \mathbf{0})}{K} \tag{19}$$

is the average number of non-zero information bits in the code words with minimum Hamming weight. Observing (18), it follows that to obtain a good ML performance, the minimum Hamming distance of the code must be large.

4 Performance Comparison

In this section, we compare the theoretical ML performance of section 3 with the performance of the iterative decoding algorithm from [3] , obtained from simulations. Figures 4 and 5 show the BER as function of E_s/N_0 for the codes with parameters summarized in table 1, for BPSK and QPSK, respectively. For low E_s/N_0, we observe that the performance of the iterative decoding algorithm is better than the performance obtained with the expression (18). This effect is caused by the union bound approximation. As expected, for low E_s/N_0, the expression (18) is not a good approximation of the actual ML performance, which in reality will be better than the performance of the iterative decoding algorithm. For higher E_s/N_0, we observe that for the codes with parameters $(K, N) = (506, 529)$ and $(K, N) = (484, 529)$, the curves corresponding to the iterative decoding algorithm and the ML decoder nearly coincide: the iterative decoding algorithm performs nearly optimal. However, for the third code ($(K, N) = (462, 529)$ with girth= 6), there exists a gap between the performance curves of the iterative decoder and the ML decoder: the iterative decoding algorithm becomes suboptimal. This effect is caused by the girth of the code. For cycle-free codes, i.e. when the girth of the code is infinite, the belief propagation method results in optimal decoding [1], [4], [8]. This can be explained as follows. In the iterative decoding algorithm, information is exchanged between the nodes - the received bits and the syndrome bits - in order to compute the marginal posterior probability that a received bit is erroneous. In [8], it is shown that for sufficiently high girths, the expected fraction of incorrectly passed messages between the nodes approaches zero when the number of iterations increases.

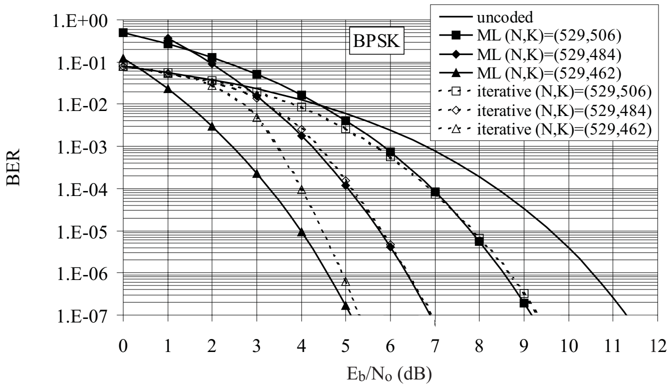

Fig. 4. BPSK performance

Table 1. Overview of the code parameters

N	K	p	k	j	$d_{H,min}$	girth
529	506	23	23	1	2	∞
529	484	23	23	2	4	8
529	462	23	23	3	6	6

Hence, for sufficiently high girths, the iterative decoding algorithm approaches the performance of the optimal ML decoder. When a code contains short cycles (low girth), the stronger coupling between the nodes gives rise to an increased fraction of incorrectly passed messages. This fraction does not converge to zero when the number of iterations increases: the iterative decoder behaves suboptimal. Hence, to obtain with the iterative decoder a performance that is close to the optimal ML performance, the girth of the code must be sufficiently high.

5 Conclusions

In this paper, we have derived the ML performance for low-density parity check codes for BPSK and QPSK transmission over a Gaussian channel. The theoretical ML performance is compared with the performance of the iterative decoding algorithm proposed in [3]. The performance of the iterative decoder turns out to be nearly optimal (close to the ML performance) when the girth of the code is sufficiently high (girth> 6). When the girth of the code is equal or smaller than 6, the iterative decoder performs suboptimal. Hence, to obtain good LDPC codes that can compete with turbo codes, the codes must have both a high minimum Hamming distance (to obtain an excellent ML performance) and a high girth (so that the iterative decoder performs nearly optimal).

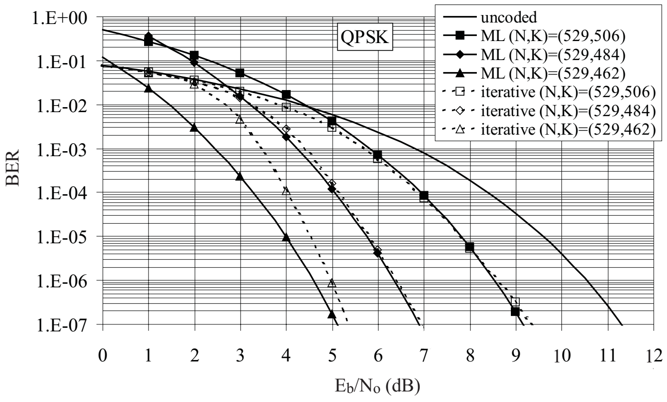

Fig. 5. QPSK performance

Acknowledgment

The first author gratefully acknowledges the financial support from the Belgian National Fund for Scientific Research (FWO Flanders).

References

1. R.G. Gallager, "Low-Density Parity-Check Codes", IRE Trans. Info. Theory, IT-8, Jan 1962, pp. 21-28
2. C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes", Proc. 1993 IEEE Int. Conference on Communications, ICC, Geneva, Switzerland, 1993, pp. 1064-1070
3. D.J.C. MacKay, "Good Error Correcting Codes Based on Very Sparse Matrices", IEEE Trans. on Information Theory, Vol 45, no 2, 1999, pp. 399-431
4. J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufman Publishers, San Francisco, 1988
5. D.J.C. MacKay, R.M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes", Electronics Letters, Vol 32, no 18, Aug 1996, pp. 1645-1646
6. S.Y. Chung, G.D. Forney, T.J. Richardson, R. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045dB of the Shannon Limit", IEEE Communications Letters, Vol 5, Feb 2001, pp. 58-60
7. T.J. Richardson, A. Shokrollahi, R. Urbanke, "Design of Capacity-Approaching Low-Density Parity-Check Codes", IEEE Trans. on Information Theory, Vol 47, Feb 2001, pp. 617-637
8. T. Richardson and R. Urbanke. "The Capacity of Low-Density Parity Check Codes under Message-Passing Decoding", submitted to IEEE Trans. on Information Theory