

Good Codes based on Very Sparse Matrices

David J.C. MacKay and Radford M. Neal

Note: the following paper was completed 6th October 1995 and was published in “Cryptography and Coding. 5th IMA Conference”, ed. Colin Boyd, Lecture Notes in Computer Science number 1025, pp. 100-111 (1995) Springer, Berlin.

Unfortunately this paper contains two errors with respect to Gallager’s work on low density parity check codes. We gained the impression from the literature that “the sparse parity check codes studied by Gallager are bad,” but this is in fact not the case. We also had the impression that Gallager’s decoding algorithm was the same as Meier and Staffelbach’s, and that our use of belief propagation was a new innovation. However, Gallager in fact proposed and used the identical belief propagation algorithm in 1962.

We became aware of these errors shortly before the IMA conference on Cryptography and Coding (December 1995). We established that Gallager’s low density parity check codes share all the ‘goodness’ properties of the ‘MN’ codes presented in this paper, and that their empirical performance is superior, as described in our more recent papers.

Good Codes based on Very Sparse Matrices

David J.C. MacKay¹ and Radford M. Neal²

¹ Cavendish Laboratory, Cambridge, CB3 0HE. United Kingdom.

² Depts. of Statistics and Computer Science, Univ. of Toronto, M5S 1A1. Canada.

Abstract. We present a new family of error-correcting codes for the binary symmetric channel. These codes are designed to encode a *sparse* source, and are defined in terms of very sparse invertible matrices, in such a way that the decoder can treat the signal and the noise symmetrically. The decoding problem involves only very sparse matrices and sparse vectors, and so is a promising candidate for practical decoding.

It can be proved that these codes are ‘very good’, in that sequences of codes exist which, when optimally decoded, achieve information rates up to the Shannon limit.

We give experimental results using a free energy minimization algorithm and a belief propagation algorithm for decoding, demonstrating practical performance superior to that of both Bose-Chaudhury-Hocquenghem codes and Reed-Muller codes over a wide range of noise levels.

We regret that lack of space prevents presentation of all our theoretical and experimental results. The full text of this paper may be found elsewhere [6].

1 Background

In 1948, Shannon [14] proved that there exist block codes, for a given memoryless channel, that achieve arbitrarily small probability of error ϵ at any communication rate R up to the capacity C of the channel. We will refer to such code families as ‘very good’ codes. By ‘good’ codes we mean code families that achieve arbitrarily small probability of error ϵ at non-zero communication rates R up to some R_{\max} that may be *less than* the capacity C of the given channel. By ‘bad’ codes we mean code families which can only achieve arbitrarily small probability of error ϵ by decreasing the information rate R to zero. (This does not mean that they are useless for practical purposes.) By ‘practical’ codes we mean code families which can be encoded and decoded in time and space polynomial in the block length.

Since 1948, few constructive and practical codes that are good have been found, fewer still that are practical, and none at all that are both practical and very good [8]. Goppa’s recent algebraic geometry codes (reviewed in [15]) appear to be both practical and good, but we believe that the literature has not established whether they are very good.

In this paper we present a new code family that we call ‘MN codes’. These codes have a very sparse structure that shows promise for practical decoding.

At the same time it can be proved that these codes are very good, in that sequences of codes exist which, when optimally decoded, achieve information rates up to the Shannon limit of the binary symmetric channel [6]. In sections 3 and 4 we describe empirical results of computer experiments using first a free energy minimization algorithm [5] and second a ‘belief propagation’ algorithm for decoding. Our experiments show that practical performance significantly superior to that of BCH and Reed-Muller codes (in terms of information rate for a given probability of decoder error) can be achieved by MN codes.

2 Description of MN codes

We will denote the error probability of the binary symmetric channel (BSC) by f_n , where $f_n < 0.5$, and the binary entropy function by $H_2(f) = f \log_2(1/f) + (1-f) \log_2(1/(1-f))$. The *weight* of a vector or matrix is the number of 1s in it. We denote the weight of a vector \mathbf{x} by $w(\mathbf{x})$. The *density* of a source of random bits is the expected fraction of 1 bits. A source is *sparse* if its density is less than 0.5. A vector \mathbf{v} is *very sparse* if its density vanishes as its length increases, for example, if a constant number t of its bits are 1s. The capacity $C(f_n)$ of a BSC with noise density f_n is, in bits per cycle, $C(f_n) = 1 - H_2(f_n)$. The rate $R_0(f_n)$ is

$$R_0(f_n) \equiv 1 - \log_2 \left[1 + 2\sqrt{f_n(1-f_n)} \right]. \quad (1)$$

This is the computational cutoff of sequential decoding for convolutional codes—the rate beyond which the expected cost of achieving vanishing error probability using sequential decoding becomes infinite.

The Gilbert bound $GV(f_n)$ is

$$GV(f_n) = \begin{cases} 1 - H_2(2f_n) & f_n < 1/4 \\ 0 & f_n \geq 1/4. \end{cases} \quad (2)$$

This is the rate at which one can communicate with a code whose codewords satisfy the Gilbert-Varshamov minimum distance bound, assuming bounded distance decoding [7].

2.1 Conventional linear codes, and the ideas behind MN codes

A linear error correcting code can be represented by a N by K binary matrix \mathbf{G} (the generator matrix), such that a binary message \mathbf{s} is encoded as the vector $\mathbf{t} = \mathbf{G}\mathbf{s} \bmod 2$ (figure 1a). (Note that our generator matrices act to the right rather than the left.) The channel adds noise \mathbf{n} to this vector with the resulting received signal \mathbf{r} being given by:

$$(\mathbf{G}\mathbf{s} + \mathbf{n}) \bmod 2 = \mathbf{r}. \quad (3)$$

The decoder’s task is to infer \mathbf{s} given the received message \mathbf{r} , and the assumed noise properties of the channel. The *optimal decoder* returns the message \mathbf{s} that

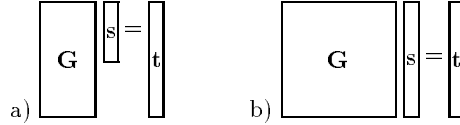


Fig. 1. a) A conventional code. The source vector \mathbf{s} , of length K , is dense. The transmitted vector \mathbf{t} is of length $N > K$. Here $N = 2K$, so the symbol rate and information rate are both $K/N = 0.5$ bits. b) Square code for a sparse source. The symbol rate is 1, but if the density of the source, f_s , is 0.1 then the information rate is $H_2(0.1) = 0.47$, about the same as that of the conventional code.

maximizes the posterior probability

$$P(\mathbf{s}|\mathbf{r}, \mathbf{G}) = \frac{P(\mathbf{r}|\mathbf{s}, \mathbf{G})P(\mathbf{s})}{P(\mathbf{r}|\mathbf{G})}. \quad (4)$$

It is often not practical to implement the optimal decoder.

It is conventional to define the error correcting code to have $N > K$, and to use signals \mathbf{s} of density $f_s = 0.5$. The $(N - K)$ extra bits are parity check bits, which produce redundancy in the transmitted vector \mathbf{t} . This redundancy is exploited by the decoding algorithm to infer the noise vector \mathbf{n} .

MN codes take a different approach. Instead of adding redundancy in the form of parity check bits, we assume that the source itself is redundant, having f_s , the density of \mathbf{s} , less than 0.5. Consecutive source symbols are independent and identically distributed. Redundant sources of this type can be produced from other sources by using a variation on arithmetic coding [16, 13]; one simply reverses the role of encoder and decoder in a standard arithmetic coder based on a model corresponding to the sparse messages [6]. Given that the source is already redundant, we are no longer constrained to have $N > K$. In MN codes, N may be less than K , equal to K or greater than K . We distinguish between the ‘symbol rate’ of the code, K/N , and the ‘information rate’ of the code, $H_2(f_s)K/N$. Error-free communication may be possible if the information rate is less than the capacity of the channel. For example, consider a BSC having $f_n = 0.1$, and assume that we have a source with density $f_s = 0.1$. Then we might construct a code with $N = K$, *i.e.*, a square linear code with symbol rate 1 (figure 1b). The information rate, 0.47, is less than the channel capacity, 0.53, so it is plausible that we might construct a sequence of codes of this form achieving vanishing probability of error.

The ideas behind MN codes are (1) that we use a sparse source and (2) that we construct the generator matrix in terms of *invertible* matrices, such that the sparse source and the sparse noise can be treated symmetrically in the decoding problem.

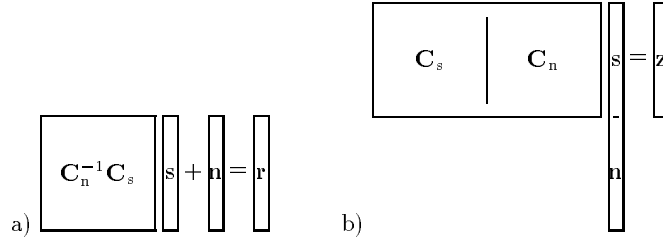


Fig. 2. Pictorial representation of MN Code with $\rho = 1$. a) Encoding, transmission and reception. b) Decoding. The matrices C_s and C_n are very sparse. The vectors s and n are sparse. The vector z is given by $z = C_n r$.

2.2 Construction of MN codes

The encoder is a linear block code constructed from very sparse matrices as follows. A transmitted block length N and a source block length $K = \rho N$ are selected. Figures 1 and 2 illustrate the case $\rho = 1$. The symbol rate of the code is ρ and the information rate is $\rho H_2(f_s)$. We select a *column weight* t , which is an integer greater than or equal to 3. We create two matrices C_n and C_s as follows.

The matrix C_n is a square $N \times N$ matrix that is very sparse and invertible. It is created randomly with exactly weight t per column and weight t per row. [Such a random sparse matrix is not necessarily invertible, but there is a probability (for large N) of about 0.29 that it is.] The inverse C_n^{-1} of this matrix is computed. This inverse is likely to be a dense matrix. The inversion takes N^3 time and is performed once only.

The matrix C_s is a rectangular $N \times K$ matrix that is very sparse. [N rows and K columns.] It is created randomly with exactly weight t per column and a weight per row as uniform as possible. If ρ is chosen to be an appropriate ratio of integers then the number per row can be constrained to be exactly ρt .

We mention three variations on this construction.

1. By slightly relaxing the constraint of weight t per column (by allowing one or two columns to have weight $t + 1$), a random very sparse C_n may easily be made invertible, by flipping one or two bits.
2. When generating the matrices C_s and C_n , one can constrain all pairs of columns in the matrix $[C_s C_n]$ to have an overlap (the number of 1s in common between the two vectors) ≤ 1 . This is expected to improve the properties of the ensemble of codes, for reasons explained in [6].
3. One can further constrain the matrix $[C_s C_n]$ so that the topology of the corresponding belief network does not contain short cycles. This is discussed further in section 3.

2.3 Encoding

A source vector \mathbf{s} of length ρN is encoded into a transmitted vector \mathbf{t} defined by (figure 2a):

$$\mathbf{t} = \mathbf{C}_n^{-1} \mathbf{C}_s \mathbf{s} \text{ mod } 2. \quad (5)$$

This encoding operation takes time of order $\min[\rho N t + N^2, \rho N^2]$.

2.4 The decoding problem

The received vector is

$$\mathbf{r} = \mathbf{t} + \mathbf{n} \text{ mod } 2, \quad (6)$$

where the noise, \mathbf{n} , is assumed to be a sparse random vector with independent identically distributed bits, density f_n . The first step of the decoding is to compute:

$$\mathbf{z} = \mathbf{C}_n \mathbf{r}, \quad (7)$$

which takes time of order Nt . Because $\mathbf{z} = \mathbf{C}_n(\mathbf{t} + \mathbf{n}) = \mathbf{C}_s \mathbf{s} + \mathbf{C}_n \mathbf{n}$, the decoding task is then to solve for $\mathbf{x} = \begin{bmatrix} \mathbf{s} \\ \mathbf{n} \end{bmatrix}$ the equation:

$$\mathbf{A} \mathbf{x} = \mathbf{z}, \quad (8)$$

where \mathbf{A} is the N by $(K+N)$ matrix $[\mathbf{C}_s \mathbf{C}_n]$ (see figure 2b). The optimal decoder, when $f_s = f_n$, is an algorithm that finds the sparsest vector $\hat{\mathbf{x}}$ that satisfies $\mathbf{A} \hat{\mathbf{x}} = \mathbf{z}$.

We emphasize two properties of equation (8):

1. There is a pleasing symmetry between the sparse source vector \mathbf{s} and the sparse noise vector \mathbf{n} , especially if $f_s = f_n$.
2. Both the matrix \mathbf{A} and the unknown vector \mathbf{x} are sparse. The vector \mathbf{x} has density f_s or f_n , and the matrix \mathbf{A} is very sparse, having only t 1s per column, where t may be much less than N . One might therefore hope that it is practical to solve this decoding problem. The decoding problem is of the type studied by Gallager [4]. However, the sparse parity check codes studied by Gallager are bad. The trick that makes MN codes good is the construction in terms of an invertible matrix.

We now describe theoretical properties that we have proved for MN codes. We then describe empirical results with a practical decoding algorithm.

2.5 Theoretical properties proven for MN codes

In [6] we prove properties of these codes by studying properties of a ‘typical set decoder’ [3] for the decoding problem $\mathbf{A} \mathbf{x} = \mathbf{z}$, averaging over an ensemble of random matrices \mathbf{A} . We prove two theorems (our proofs are computer-aided), whose implications are as follows.

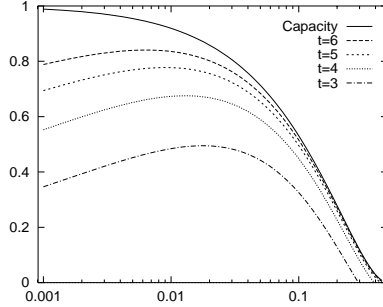


Fig. 3. Main theoretical result. Lower bounds $R^*(f, t)$ on achievable information rate versus noise level f for MN codes with t from 3 to 6. In bits, compared with the channel capacity. The lines are lower bounds on rates achievable by MN codes. As the weight per column t increases the achievable region rises towards the fundamental limit, the capacity.

1. MN codes with weight per column $t \geq 3$ are *good*, *i.e.*, can achieve error-free transmission up to a non-zero information rate $R^*(f, t)$, if N is made sufficiently large. This rate is plotted numerically in figure 3. This rate is less than the capacity $C(f)$, but for useful values of f , even for t as small as 4, it is not much below the capacity.
2. MN codes are *very good*—if we are allowed to choose t , then we can get arbitrarily close to capacity, still using very sparse matrices with t/N arbitrarily small. The second theorem states:

Given a density $f < 0.5$, a desired information rate $R < C(f)$, and a desired block error probability $\epsilon > 0$, there exists an integer $t \geq 3$, a symbol rate ρ and an N_{\min} such that for any $N > N_{\min}$, there is a matrix \mathbf{A} having N rows and $K' = N + K = (\rho + 1)N$ columns with weight t or less per column, with the following property: if \mathbf{x} has density f then the optimal decoder from $\mathbf{z} = \mathbf{A}\mathbf{x}$ back to $\hat{\mathbf{x}}$ achieves a probability of error less than ϵ , and the information rate that is achieved is $\geq R$.

3 Practical decoding by free energy minimization

We generated random matrices \mathbf{A} corresponding to symbol rate $\rho = 1$, with uniform weight $t = 4$ per column and $t_r = 8$ per row. We first attempted to solve the decoding problem using a variational free energy minimization algorithm [5]. We found that as the block size N was increased at a constant information rate, the performance improved.

We examined the errors made by the free energy minimization decoder and found that they tended to occur when the vector \mathbf{x} was such that another slightly

different typical vector \mathbf{x}' had a similar (but not identical) encoding \mathbf{z}' . These errors were attributable to rare topologies in the network corresponding to the \mathbf{A} matrix such as the topology illustrated in figure 4c. We can eliminate the possibility of these errors by modifying the ensemble of random matrices \mathbf{A} so that the corresponding network does not have short cycles in it.

The topological modifications gave codes which were able to communicate at higher rates with a smaller probability of error. The conclusion of these experiments was that MN codes, when decoded by free energy minimization, can be superior to Reed-Muller codes, but not to BCH codes. Significantly better results were obtained when we used the belief net decoder which we now describe.

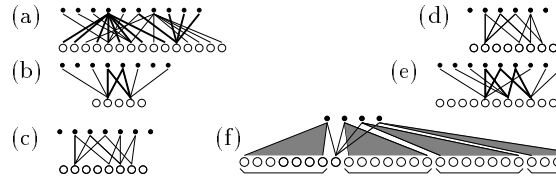


Fig. 4. The vectors \mathbf{x} and \mathbf{z} viewed as nodes in a belief network. White circles denote bits x_k . Black dots denote checks z_n . We illustrate the case $t = 4, t_r = 8$. (a) This figure emphasizes with bold lines the 8 connections to one check and the 4 connections from one bit. Every bit x_k is the parent of 4 checks z_n , and each check z_n is the child of 8 bits. (b-e) Certain topological structures are undesirable in the network defined by the matrix \mathbf{A} : in (b) there is a cycle of length 4 in the network; we forbid this topology by saying, equivalently, that the overlap between two columns of \mathbf{A} must not exceed 1; in (c, d, e) more complex topologies are illustrated. Our most successful experiments have used matrices \mathbf{A} in which these topologies are also forbidden [we eliminate bits that are involved in structures like the ‘doublet’ (e), of which (c) and (d) are hazardous special cases]. This means that every bit’s ‘friends’ (other bits that are parents of its children) consist of t non-overlapping sets of bits as shown in (f).

4 Belief network decoding

We have developed a ‘belief net decoder’ for the problem $\mathbf{Ax} = \mathbf{z} \bmod 2$, which generalizes the methods of Gallager [4] and Meier and Staffelbach [9] by using methods of belief propagation over networks [11].

We refer to the elements z_n corresponding to each row $n = 1 \dots N$ of \mathbf{A} as checks. We think of the set of bits \mathbf{x} and checks \mathbf{z} as making up a ‘belief network’, also known as a ‘Bayesian network’, ‘causal network’, or ‘influence diagram’, in which every bit x_k is the parent of t checks z_n , and each check z_n is the child of t_r bits (figure 4). We aim, given the observed checks, to compute the marginal posterior probabilities $P(x_k = 1 | \mathbf{z}, \mathbf{A})$ for each k . Algorithms for the computation of such marginal probabilities in belief networks are found in [11]. These

computations are expected to be intractable for the belief net corresponding to our problem $\mathbf{Ax} = \mathbf{z} \bmod 2$ because its topology contains many cycles. However, it is interesting to implement the decoding algorithm that would be appropriate if there were no cycles, on the assumption that the errors introduced might be relatively small (c.f. [1]). As the size N of the code is increased, it becomes increasingly easy to produce codes in which there are no cycles of any given length, so we expect that, asymptotically, this algorithm will be an effective algorithm.

4.1 The algorithm

In the following algorithm quantities q_{nk} and r_{nk} associated with each 1 bit in the \mathbf{A} matrix are iteratively updated. We denote the set of bits k that participate in check n by $\mathcal{K}(n) \equiv \{k : A_{nk} = 1\}$. Similarly we define the set of checks in which bit k participates, $\mathcal{N}(k) \equiv \{n : A_{nk} = 1\}$.

Initialization. Let $p_k^0 = P(x_k = 0)$ (the prior probability that bit x_k is 0), and let $p_k^1 = P(x_k = 1) = 1 - p_k^0$. Normally, p_k^1 will be either f_s or f_n , depending on whether bit k is part of the message or the noise. For every (k, n) such that $A_{nk} = 1$ the variables q_{nk}^0 and q_{nk}^1 are initialized to the values p_k^0 and p_k^1 respectively.

Horizontal pass. In the horizontal step of the algorithm, we run through the checks n and compute for each $k \in \mathcal{K}(n)$ two probabilities: the probability of the observed value of z_n arising when $x_k = 0$, given that the other bits $\{x_{k'}, k' \neq k\}$ have a separable distribution given by the probabilities $\{q_{nk'}^0, q_{nk'}^1\}$:

$$r_{nk}^0 = \sum_{\{x_{k'} : k' \neq k\}} P(z_n \mid x_k = 0, \{x_{k'} : k' \neq k\}) \prod_{k' \neq k} q_{nk'}^{x_{k'}} \quad (9)$$

and the probability of the observed value of z_n arising when $x_k = 1$, r_{nk}^1 , defined similarly. These probabilities can be computed efficiently using forward and backward passes (c.f. [5]), in which products of the differences $\delta q_{nk} \equiv q_{nk}^0 - q_{nk}^1$ are computed. We obtain $\delta r_{nk} \equiv r_{nk}^0 - r_{nk}^1$ from the identity:

$$\delta r_{nk} = (-1)^{z_n} \prod_{k' \in \mathcal{K}(n), k' \neq k} \delta q_{nk'}. \quad (10)$$

Vertical pass. The vertical step takes the computed values of r_{nk}^0 and r_{nk}^1 and updates the values of the probabilities q_{nk}^0 and q_{nk}^1 . For each k we compute:

$$q_{nk}^0 = \alpha_{nk} p_k^0 \prod_{n' \in \mathcal{N}(k), n' \neq n} r_{n'k}^0, \quad q_{nk}^1 = \alpha_{nk} p_k^1 \prod_{n' \in \mathcal{N}(k), n' \neq n} r_{n'k}^1, \quad (11)$$

where α_{nk} is a constant such that $q_{nk}^0 + q_{nk}^1 = 1$. We can also compute the ‘pseudoposterior probabilities’ q_k^0 and q_k^1 at this iteration, given by:

$$q_k^0 = \alpha_k p_k^0 \prod_{n \in \mathcal{N}(k)} r_{nk}^0, \quad q_k^1 = \alpha_k p_k^1 \prod_{n \in \mathcal{N}(k)} r_{nk}^1. \quad (12)$$

At this point, the algorithm repeats from the horizontal pass.

Decoding. Our decoding procedure is to set \hat{x}_k to 1 if $q_k^1 > 0.5$ and see if the checks $\mathbf{A}\hat{\mathbf{x}} = \mathbf{z}$ are all satisfied, halting when they are, and declaring a failure if some maximum number of iterations (*e.g.*, 1000) occurs without successful decoding.

4.2 Relationship to Gallager’s algorithm

Gallager [4] and Meier and Staffelbach [9] implemented algorithms very similar to this belief net decoder, also studied by Mihaljević and Golić [10]. The main difference in their algorithms is that they did not distinguish between the probabilities q_{nk}^0 and q_{nk}^1 for different values of n ; rather, they computed q_k^0 and q_k^1 , as given above, and then proceeded with the horizontal pass with all q_{nk}^0 set to q_k^0 and all q_{nk}^1 set to q_k^1 .

4.3 Empirical results: belief net decoder

We found the performance of the belief net decoder to be far better than that of the free energy minimization decoder. We found that the results were best for $t = 3$ and became steadily worse as t increased.

In figure 5 we compare two MN codes with BCH codes, which are described in [12] as “the best known constructive codes” for memoryless noisy channels, and with Reed-Muller (RM) codes (block sizes up to 1024). Figure 5 shows the codes’ probability of block error versus their rate. All relevant BCH codes listed in [12] are included. To compute the probability of error for BCH codes we evaluated the probability of more than t errors in n bits, as specified in the (n, k, t) description of the code. In principle, it may be possible in some cases to make a BCH decoder that corrects more than t errors, but according to Berlekamp [2], “little is known about... how to go about finding the solutions” and “if there are more than $t + 1$ errors then the situation gets very complicated very quickly.” Similarly, for RM codes of minimum distance d , performance was computed assuming that more than $\lfloor d/2 \rfloor$ errors cannot be corrected.

The mean number of iterations of the algorithm to obtain a successful decoding was about 20 for all the experiments reported here. In some cases as many as 800 iterations took place before a successful decoding emerged.

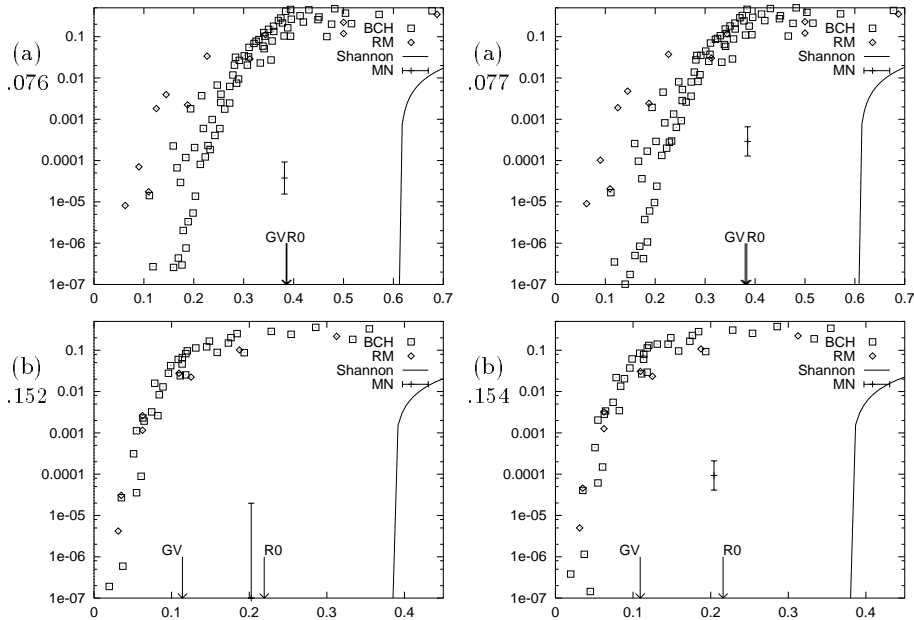


Fig. 5. Comparison of empirical decoding results for MN codes using belief network decoder with calculated performance of Reed-Muller codes and BCH codes, and the Shannon limit. A BSC with (a) $f_n = 0.076-7$ (b) $f_n = 0.152-4$ is assumed. Horizontal axis: information rate R . Vertical axis: block error probability. The best codes are towards the bottom (low error probability) and the right (large rate). Curve: Shannon limit on achievable (rate, bit error probability) values. Arrows show the values of R_0 and $GV(f_n)$ for this channel. Diamonds: Reed-Muller codes. Boxes: BCH codes. MN codes: Empirical results shown are for two topologically modified matrices with $N = 10000$ rows and $N+K$ columns where $K =$ (a) 9839 (b) 3296. The weight per column was $t = 3$. In the case where the error bars extend down to the bottom of the error probability axis, no decoding errors occurred in more than 100,000 trials.

5 Discussion

Our experiments have demonstrated excellent error correction at rates well above the Gilbert bound. In [6] we give an analysis of two practical decoding algorithms. This analysis, and the empirical results we have described, lead us to conjecture that *given a BSC with noise density f , there exist practical decoders for MN codes with any rate R up to $R_0(f)$ which can achieve negligible probability of error, for sufficiently large N .*

The properties of MN codes that we have demonstrated appear to constitute a significant step forward in information theory and coding theory.

The descriptive complexity of these codes is $t(N+K) \log N$, which is much smaller than the complexity of arbitrary linear codes. The set-up time for the

code scales as N^3 , the encoding time as N^2 , and the decoding time as N , where N is the block size.

5.1 Contrasts with convention in coding theory

In a conventional linear (N, K) code, the codewords form a complete linear subspace of $\{0, 1\}^N$. MN codes are only linear in the sense that the transmitted vector \mathbf{t} is a linear function of a source vector \mathbf{s} . The source is *sparse*, so the codewords that have high probability are only a small subset of a complete linear subspace.

We have obtained the biggest improvement over BCH codes and RM codes by going to high noise levels, *e.g.*, $f_n = 0.15$. Critics might assert that real channels do not have such high noise levels. We would respond that perhaps they ought to—if one increases the clock rate of a channel so that its noise level also increases, there might well be a net increase in capacity. Maybe the main reason that channels with high noise levels are not used is that until now the available codes for error correction have not been good enough.

5.2 Future work

MN codes can also be defined over q -ary alphabets consisting of the elements of $GF(q)$. These codes would be suitable for the q -ary symmetric channel. The decoding algorithms presented here would also generalize. It remains to be established whether our decoders' performance would be any better or worse under this generalization to q -ary alphabets.

We conjecture that as we get closer to the Shannon limit, the decoding problem gets harder. It would be interesting to obtain a convergence proof for the belief net decoding algorithm and to develop ways of reducing the inaccuracies introduced by the approach of ignoring the cycles present in the belief network. The most interesting challenge is to understand whether $R_0(f)$ is indeed the fundamental limit for practical decoding of MN codes.

Acknowledgements

DJCM (mackay@mrao.cam.ac.uk) is grateful to Roger Sewell and David Aldous for helpful discussions and M.D. MacLeod and the Computer Laboratory, Cambridge for kind loans of books. DJCM also thanks Geoff Hinton for generously supporting his visits to the University of Toronto. DJCM was supported by the Royal Society Smithson research fellowship. RMN (radford@stat.toronto.edu) was assisted by a grant from the Natural Sciences and Engineering Research Council of Canada.

References

1. S. Andreassen, M. Woldbye, B. Falck, and S. Andersen. MUNIN - a causal probabilistic network for the interpretation of electromyographic findings. In *Proc. of the 10th National Conf. on AI, AAAI: Menlo Park CA.*, pages 121–123, 1987.

2. E. R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, New York, 1968.
3. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.
4. R. G. Gallager. Low density parity check codes. *IRE Trans. Info. Theory*, IT-8:21–28, Jan 1962.
5. D. J. C. MacKay. Free energy minimization algorithm for decoding and cryptanalysis. *Electronics Letters*, 31(6):446–447, 1995.
6. D. J. C. MacKay and R. M. Neal. Good codes based on very sparse matrices. Available from <http://131.111.48.24/>, 1995.
7. F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. North-Holland, Amsterdam, 1977.
8. R. J. McEliece. *The theory of information and coding: a mathematical framework for communication*. Addison-Wesley, Reading, Mass., 1977.
9. W. Meier and O. Staffelbach. Fast correlation attacks on certain stream ciphers. *J. Cryptology*, 1:159–176, 1989.
10. M. J. Mihaljević and J. D. Golić. Convergence of a Bayesian iterative error-correction procedure on a noisy shift register sequence. In *Advances in Cryptology - EUROCRYPT 92*, volume 658, pages 124–137. Springer-Verlag, 1993.
11. J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Mateo, 1988.
12. W. W. Peterson and E. J. Weldon, Jr. *Error-Correcting Codes*. MIT Press, Cambridge, Massachusetts, 2nd edition, 1972.
13. J. Rissanen and G. G. Langdon. Arithmetic coding. *IBM Journal of Research and Development*, 23:149–162, 1979.
14. C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:379–423, 623–656, 1948.
15. M. A. Tsfasman. Algebraic-geometric codes and asymptotic problems. *Discrete Applied Mathematics*, 33(1-3):241–256, 1991.
16. I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.