Good Error-Correcting Codes Based on Very Sparse Matrices

David J. C. MacKay

Abstract— **We study two families of error-correcting codes defined in terms of very sparse matrices. "MN" (MacKay–Neal) codes are recently invented, and "Gallager codes" were first investigated in 1962, but appear to have been largely forgotten, in spite of their excellent properties. The decoding of both codes can be tackled with a practical sum-product algorithm.**

We prove that these codes are "very good," in that sequences of codes exist which, when optimally decoded, achieve information rates up to the Shannon limit. This result holds not only for the binary-symmetric channel but also for any channel with symmetric stationary ergodic noise.

We give experimental results for binary-symmetric channels and Gaussian channels demonstrating that practical performance substantially better than that of standard convolutional and concatenated codes can be achieved; indeed, the performance of Gallager codes is almost as close to the Shannon limit as that of turbo codes.

Index Terms— **Error-correction codes, iterative probabilistic decoding, low-complexity decoding, nonlinear codes, Shannon limit.**

I. Introduction

A. Background

In 1948, Shannon [58] proved that for any channel there exist families of block codes that achieve arbitrarily small probability of error at any communication rate up to the capacity 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 at nonzero communication rates up to some maximum rate that may be *less than* the capacity of the given channel. By "bad" codes we mean code families that can only achieve arbitrarily small probability of error by decreasing the information rate to zero. (Bad codes are not necessarily useless for practical purposes.) By "practical" codes we mean code families which can be encoded and decoded in time and space polynomial in the blocklength.

Shannon's proof was nonconstructive and employed random codes for which there is no practical encoding or decoding algorithm. Since 1948, it has been proved that there exist very good cyclic codes (nonconstructively) [45], and that very good codes with a short description in terms of permutations can be produced [1]; and an explicit algebraic construction of very good codes for certain channels was given in 1982 [19]. But no practical decoding algorithm is known for any of these codes, and it is known that the general linear decoding problem (find the maximum-likelihood source vector $\boldsymbol{s}$ in the equation $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{s} + \boldsymbol{n} = \boldsymbol{r} \bmod 2$, where $\boldsymbol{G}$ is a generator matrix, $\boldsymbol{n}$ is a noise vector, and $\boldsymbol{r}$ is the received vector) is NP-complete [10]. Convolutional codes (which can be viewed as block codes with memory) can approach the Shannon limit as their constraint length increases but the complexity of their best known decoding algorithms grows exponentially with the constraint length. For a long time a generally held view was that for practical purposes a channel's effective capacity was a rate "R_0" which is smaller than the Shannon capacity, if convolutional codes were used; and many believed this conjecture applied to all codes, speculating that practical communication beyond R_0 was impossible. Forney proved that there do exist very good "concatenated" codes that are practical [23]; but the proof was also nonconstructive [45].

When it comes to practical, *constructive* codes, constructions have been demonstrated of codes based on concatenation that are good, though not very good, but most known practical codes are asymptotically bad [45]. Goppa's algebraic-geometry codes, reviewed in [66], appear to be both practical and good (with practical decoding proven possible up to the Gilbert bound), but we believe that the literature has not established whether they are very good. The best practical decoding algorithm that is known for these codes [22] appears to be prohibitively costly (N^3) to implement, and algebraic-geometry codes do not appear to be destined for practical use.

Thus the conventional view is that there are few known constructive codes that are good, fewer still that are practical, and none at all that are both practical and very good. It seems to be widely believed that while almost any random linear code is good, codes with structure that allows practical coding are likely to be bad [45], [15]. Battail expresses an alternative view, however, that "we can think of good codes, and we can decode them" [6]. This statement is supported by the results of the present paper.

In this paper we study the theoretical and practical properties of two code families. Gallager's low-density parity-check codes are defined in terms of a very sparse random parity-check matrix [26], [27], [41]. "MN codes" are also defined in terms of very sparse random matrices, and were first presented in [40]. (MN stands for MacKay–Neal; MacKay and Neal generalized MN codes to Gallager codes, then realized

Manuscript received June 29, 1997; revised July 27, 1998. This work was supported by the Royal Society Smithson research fellowship and by the Gatsby charitable foundation.

The author is with Cavendish Laboratory, Cambridge, CB3 0HE, U.K. (e-mail: mackay@mrao.cam.ac.uk).

Communicated by A. Vardy, Associate Editor for Coding Theory.

Publisher Item Identifier S 0018-9448(99)01395-4.

that they had rediscovered Gallager's work.) MN codes are unconventional in that redundancy can be incorporated in the transmitted codewords not only by using a $K \times N$ generator matrix with transmitted blocklength N greater than the source blocklength K, but also by using a source that is itself redundant.

These code families both have two important properties. First, because the codes are constructed from sparse matrices, they have simple and practical decoding algorithms which work, empirically, at good communication rates. Second, we prove that in spite of their simple construction these codes are *very good*—that is, sequences of codes exist which, when optimally decoded, achieve information rates up to the Shannon limit of the binary-symmetric channel. We further prove that the same codes are in fact good for any ergodic symmetric channel. Our proof may be viewed as a semiconstructive proof of Shannon's noisy channel coding theorem (semiconstructive in the sense that, while the proof still relies on an average over a set of codes, the set of codes in question is unusually small). It is indeed easy to think of good codes.

B. Definitions

For a glossary of symbols used in this paper, please see Table III.

A binary variable will be termed a *bit*. The unit of information content of a random bit with $p_0 = p_1 = 0.5$ will be termed the *shannon*. The input and output alphabets of the binary-symmetric channel (BSC) will be denoted $\{0, 1\}$. We will denote the error probability of the binary-symmetric channel by $f_{\rm n}$, where $f_{\rm n} < 0.5$.

Definition 1: The binary entropy functions $H_2(f)$ and $H_2^e(f)$ are

$$H_2(f) = f \log_2(1/f) + (1-f)\log_2(1/(1-f)) \quad (1)$$

$$H_2^e(f) = f \log_e(1/f) + (1-f)\log_e(1/(1-f)). \quad (2)$$

We will write natural logarithms as $\log(x) \equiv \log_e(x)$.

Definition 2: The *weight* of a binary vector or matrix is the number of 1's in it. The *overlap* between two vectors is the number of 1's in common between them. 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 $\boldsymbol{v}$ is *very sparse* if its density vanishes as its length increases, for example, if a constant number t of its bits are 1's.

Definition 3: A code with blocklength N and rate R satisfies the *Gilbert–Varshamov minimum-distance bound* if the minimum distance d between its codewords satisfies

$$R = 1 - H_2(d/N). \quad (3)$$

Definition 4: The *capacity* $C(f_{\rm n})$ of a binary-symmetric channel with noise density $f_{\rm n}$ is, in shannons per channel use,

$$C(f_{\rm n}) = 1 - H_2(f_{\rm n}). \quad (4)$$

The *computational cutoff rate* $R_0(f_{\rm n})$ is

$$R_0(f_{\rm n}) \equiv 1 - \log_2[1 + 2\sqrt{f_{\rm n}(1-f_{\rm n})}]. \quad (5)$$

This is the rate beyond which the expected computational cost of decoding a convolutional code with vanishing error probability using sequential decoding becomes infinite.

The *Gilbert bound* $GV(f_{\rm n})$ is

$$GV(f_{\rm n}) = \begin{cases} 1 - H_2(2f_{\rm n}), & f_{\rm n} < 1/4 \\ 0, & f_{\rm n} \geq 1/4. \end{cases} \quad (6)$$

This is the maximum rate at which one can communicate with a code which satisfies the Gilbert–Varshamov minimum-distance bound, assuming bounded distance decoding [43].

Definition 5: A model that defines a probability distribution over strings $\boldsymbol{x}$ of any length N, $P(\boldsymbol{x} \mid N)$, has *mean entropy* H_x if for any $\epsilon > 0$ and $\eta > 0$ there exists an N^* such that for all $N > N^*$

$$P\left(\left| \frac{1}{N} \log_2 \frac{1}{P(\boldsymbol{x} \mid N)} - H_x \right| > \eta \right) < \epsilon. \quad (7)$$

For example, a memoryless binary-symmetric channel's noise has mean entropy $H_{\rm n} = H_2(f_{\rm n})$, where $f_{\rm n}$ is the density of the noise; the proof of this statement, by the law of large numbers, is well known [16]. We will prove that the codes presented in this paper are good codes not only for the binary-symmetric channel but also for a wide class of channels with memory.

Definition 6: A *binary channel with symmetric stationary ergodic noise* is a binary-input, binary-output channel whose output in response to a transmitted binary vector $\boldsymbol{t}$ is given by $\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{n} \bmod 2$, where $\boldsymbol{n}$, the noise vector, has a probability distribution that is a) independent of $\boldsymbol{t}$ and b) stationary and ergodic.

For example, burst noise might be modeled by a stationary and ergodic Markov process. Such a process has a mean entropy, though the evaluation of this quantity may be challenging. The standard Gaussian channel with binary inputs is also equivalent to a binary channel with stationary ergodic noise.

We will concentrate on the case of a binary channel with symmetric noise (see Definition 6) in the body of this paper. Channels like the Gaussian channel whose inputs are binary and whose outputs are in some more general alphabet are addressed in Appendix A.

1) Linear Codes: A linear error-correcting code can be represented by an N by K binary matrix $\boldsymbol{G}^{\rm T}$ (the **generator matrix**), such that a K-bit binary message $\boldsymbol{s}$ is encoded as the N-bit vector $\boldsymbol{t} = \boldsymbol{G}^{\rm T}\boldsymbol{s} \bmod 2$. (Note that we have chosen to use column vectors so the generator matrices act to the right rather than the left.) The generator matrix is in *systematic form* if it can be written as

$$\boldsymbol{G}^{\rm T} = \begin{bmatrix} \boldsymbol{I}_K \\ \boldsymbol{P} \end{bmatrix} \quad (8)$$

where $\boldsymbol{I}_K$ is the $K \times K$ identity matrix, and $\boldsymbol{P}$ is a binary matrix. The channel adds noise $\boldsymbol{n}$ to the vector $\boldsymbol{t}$ with the resulting received signal $\boldsymbol{r}$ being given by

$$\boldsymbol{r} = (\boldsymbol{G}^{\rm T}\boldsymbol{s} + \boldsymbol{n}) \bmod 2. \quad (9)$$

The decoder's task is to infer $\boldsymbol{s}$ given the received message $\boldsymbol{r}$, and the assumed noise properties of the channel. The *optimal decoder* returns the message $\boldsymbol{s}$ that maximizes the posterior probability

$$P(\boldsymbol{s}\,|\,\boldsymbol{r},\boldsymbol{G}) = \frac{P(\boldsymbol{r}\,|\,\boldsymbol{s},\boldsymbol{G})P(\boldsymbol{s})}{P(\boldsymbol{r}\,|\,\boldsymbol{G})}. \tag{10}$$

It is often not practical to implement the optimal decoder; indeed, the general decoding problem is known to be NP-complete [10].

If the prior probability of $\boldsymbol{s}$ is assumed uniform, and the probability of $\boldsymbol{n}$ is assumed to be independent of $\boldsymbol{s}$ (c.f. Definition 6), then it is convenient to introduce the $(N-K)\times N$ **parity-check matrix** $\boldsymbol{H}$, which in systematic form is $[\boldsymbol{P}|\boldsymbol{I}_{N-K}]$. The parity-check matrix has the property $\boldsymbol{H}\boldsymbol{G}^{\mathrm{T}} = 0 \bmod 2$, so that, applying $\boldsymbol{H}$ to (9)

$$\boldsymbol{H}\boldsymbol{n} = \boldsymbol{H}\boldsymbol{r} \bmod 2. \tag{11}$$

Any other $(N-K)\times N$ matrix $\boldsymbol{A}$ whose rows span the same space as $\boldsymbol{H}$ is a valid parity-check matrix.

The decoding problem thus reduces, given the above assumptions, to the task of finding the most probable noise vector $\boldsymbol{n}$ such that

$$\boldsymbol{H}\boldsymbol{n} \bmod 2 = \boldsymbol{z} \tag{12}$$

where the syndrome vector $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{r} \bmod 2$.

C. Description of the Two Code Families

We define two code families. We explain the more conventional Gallager codes first.

1) Construction of Gallager Codes: A Gallager code is a code which has a very sparse random parity-check matrix. (Very sparse, but *not* systematic.) The parity-check matrix $\boldsymbol{A}$ can be constructed as follows. We will describe variations on this construction later.

A transmitted blocklength N and a source blocklength K are selected. We define $M = N-K$ to be the number of parity checks. We select a *column weight t*, which will initially be an integer greater than or equal to 3. We create a rectangular $M \times N$ matrix (M rows and N columns) $\boldsymbol{A}$ at random with exactly weight t per column and a weight per row as uniform as possible. If N/M is chosen to be an appropriate ratio of integers then the number per row can be constrained to be exactly tN/M; in this case, we call the resulting code a *regular* Gallager code because the bipartite graph defined by the parity-check matrix is regular.

We then use Gaussian elimination and reordering of columns to derive an equivalent parity-check matrix in systematic form $\boldsymbol{H} = [\boldsymbol{P}|\boldsymbol{I}_M]$. There is a possibility that the rows of the original matrix $\boldsymbol{A}$ are not independent (though for odd t, this has small probability); in this case, $\boldsymbol{A}$ is a parity-check matrix for a code with the same N and with smaller M, that is, a code with *greater* rate than assumed in the following sections. Redefining $\boldsymbol{A}$ to be the original matrix with its columns reordered as in the Gaussian elimination, we have the following situation.

The matrix $\boldsymbol{A} = [\boldsymbol{C}_1|\boldsymbol{C}_2]$ is composed of two very sparse matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ as follows.

The matrix $\boldsymbol{C}_2$ is a square $M\times M$ matrix that is very sparse and invertible. The inverse $\boldsymbol{C}_2^{-1}$ of this matrix in modulo 2 arithmetic has been computed during the Gaussian elimination which produced the matrix $\boldsymbol{P} = \boldsymbol{C}_2^{-1}\boldsymbol{C}_1$. (The expression $\boldsymbol{P} = \boldsymbol{C}_2^{-1}\boldsymbol{C}_1$ is the product (modulo 2) of the two matrices $\boldsymbol{C}_2^{-1}$ and $\boldsymbol{C}_1$.) The inversion takes order M^2N time and is performed once only.

The matrix $\boldsymbol{C}_1$ is a rectangular $M \times K$ matrix that is very sparse.

Encoding. We define the generator matrix of the Gallager code to be

$$\boldsymbol{G}^{\mathrm{T}} = \begin{bmatrix}\boldsymbol{I}_K\\ \boldsymbol{P}\end{bmatrix} = \begin{bmatrix}\boldsymbol{I}_K\\ \boldsymbol{C}_2^{-1}\boldsymbol{C}_1\end{bmatrix} \tag{13}$$

where $\boldsymbol{I}_K$ is the $K \times K$ identity matrix.

2) Variations:

a) When generating the matrix $\boldsymbol{A}$, one can constrain all pairs of columns in the matrix to have an overlap ≤ 1. This is expected to improve the properties of the ensemble of codes, for reasons that will become apparent in Section II-C.

b) One can further constrain the matrix $\boldsymbol{A}$ so that the topology of the corresponding bipartite graph does not contain short cycles. This is discussed further in Section IV-C.

3) The Decoding Problem for Gallager Codes: A source vector $\boldsymbol{s}$ of length K is encoded into a transmitted vector $\boldsymbol{t}$ defined by $\boldsymbol{t} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{s} \bmod 2$. If a systematic generator matrix has been computed explicitly (which takes M^2N time) then the transmitted vector can be computed by explicit multiplication in MK time. However, encoding might be possible in less time using sparse matrix methods.

The received vector is $\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{n} \bmod 2$, where the noise is $\boldsymbol{n}$. In the case of a binary-symmetric channel, $\boldsymbol{n}$ is assumed to be a sparse random vector with independent and identically distributed bits of density f_{n}. We will discuss more general channels later.

By construction, $\boldsymbol{A}$ is a parity-check matrix for $\boldsymbol{G}$—that is, $\boldsymbol{A}\boldsymbol{G}^{\mathrm{T}} = 0 \bmod 2$—so the decoding problem is to recover $\boldsymbol{t}$ by finding the most probable $\boldsymbol{n}$ that satisfies the equation

$$\boldsymbol{A}\boldsymbol{n} = \boldsymbol{z} \bmod 2 \tag{14}$$

where $\boldsymbol{z}$ is the syndrome vector $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{r} \bmod 2$, computation of which takes time of order Nt, if the sparseness of $\boldsymbol{A}$ is exploited.

The optimal decoder, in the case of a binary-symmetric channel, is an algorithm that finds the sparsest vector $\hat{\boldsymbol{n}}$ that satisfies $\boldsymbol{A}\hat{\boldsymbol{n}} = \boldsymbol{z} \bmod 2$. From $\hat{\boldsymbol{n}}$ we obtain our guess for the transmitted signal $\hat{\boldsymbol{t}} = \boldsymbol{r} + \hat{\boldsymbol{n}} \bmod 2$, the first K bits of which are the optimal guess $\hat{\boldsymbol{s}}$ for the original message.

Both the matrix $\boldsymbol{A}$ and the unknown vector $\boldsymbol{n}$ are sparse. One might therefore hope that it is practical to solve this decoding problem (though perhaps not right up to the theoretical limits of the optimal decoder). We will demonstrate a practical decoder later.

4) Construction of MN Codes: Having created the matrices $\boldsymbol{C}_2^{-1}$ and $\boldsymbol{C}_1$, we can define the generator matrix $\boldsymbol{G}$ of an MN code by the nonsystematic matrix $\boldsymbol{G}^{\mathrm{T}} = \boldsymbol{C}_2^{-1}\boldsymbol{C}_1$. The novel idea behind MN codes is that we can *constrain the source vectors to be sparse* and exploit this unconventional form of redundancy in the decoder [40]. We will discuss properties and possible applications of MN codes in Section VI. As explained in that section, the decoding of these codes involves a problem similar to the Gallager codes' decoding problem (14).

5) Overview: The *theoretical* effectiveness of Gallager and MN codes as error-correcting codes depends on the properties of very sparse matrices $\boldsymbol{A}$ in relation to the solvability of the decoding problem (14). We address the question, "how well would these codes work if we had the best possible algorithm for solving the decoding problem?"

The *practical* effectiveness of Gallager and MN codes depends on our finding a practical algorithm for solving (14) that is close enough to the optimal decoder that the desirable theoretical properties are not lost.

We show theoretically in Section II that there exist Gallager and MN codes for which the optimal decoder would achieve information rates arbitrarily close to the Shannon limit for a wide variety of channels. In Section III we present a "sum-product" decoding algorithm for Gallager codes and MN codes, first used by Gallager [26]. We give an analysis of the decoding algorithm in Section III-C. These results lead us to conjecture that there exist Gallager and MN codes which are not only good but which also achieve error rates approaching zero at a nonzero information rate when decoded using a practical algorithm. In Section IV we describe empirical results of computer experiments using the sum-product algorithm to decode Gallager codes. Our experiments show that practical performance significantly superior to that of textbook codes can be achieved by these codes on both binary-symmetric channels and Gaussian channels. In Section V we give a pictorial demonstration of the iterative decoding algorithm for a couple of Gallager codes. In Section VI we present MN codes and give theoretical and experimental results for them.

II. Limits of Optimal Decoding

We prove properties of Gallager and MN codes by studying properties of the decoding problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \bmod 2$ where the unknown vector $\boldsymbol{x}$ is sparse and $\boldsymbol{A}$ is very sparse. We make use of two standard tools: we prove properties of the optimal decoder by proving properties of a slightly suboptimal "typical set decoder" which is easier to analyze; and we average the performance of this decoder over an ensemble of very sparse matrices $\boldsymbol{A}$. A "good" average performance proves that there exist "good" matrices $\boldsymbol{A}$—indeed, that any random matrix $\boldsymbol{A}$ from the ensemble is likely to be "good." As in all proofs of goodness of coding systems, we employ a blocklength that can be increased to a sufficiently large value that an error probability smaller than a desired ϵ is achieved. To prove that Gallager and MN codes are *very* good we will also increase the weight per column, t, of the matrix $\boldsymbol{A}$, but only in such a way as to keep the matrix very sparse, i.e., $t/M \rightarrow 0$.

Previous work on low-density parity-check codes has already established some good properties of Gallager codes. Gallager [26], [27] proved that his codes have good distance properties. Zyablov and Pinsker [73] proved that Gallager codes are good and gave a practical decoder, but only for communication rates substantially below the Gilbert bound. Our approach in terms of an ideal decoder allows us to prove that the codes are good not only for the binary-symmetric channel but also for arbitrary ergodic symmetric channel models; we also prove that Gallager codes are *very* good, a result not explicitly proven in [26], [27], and [73].

A. Ensembles of Very Sparse Matrices

The properties that we prove depend on the ensemble of matrices $\boldsymbol{A}$ that is averaged over. We find it easiest to prove the desired properties by weakening the ensemble of matrices from that described in Section I-C. We introduce the following ensembles which we believe are ordered such that the later ensembles define Gallager and MN codes that have smaller average probability of error, though we do not have a proof of this statement.

1) Matrix $\boldsymbol{A}$ generated by starting from an all-zero matrix and randomly flipping t not necessarily distinct bits in each column.
2) Matrix $\boldsymbol{A}$ generated by randomly creating weight t columns.
3) Matrix $\boldsymbol{A}$ generated with weight t per column and (as near as possible) uniform weight per row.
4) Matrix $\boldsymbol{A}$ generated with weight t per column and uniform weight per row, and no two columns having overlap greater than 1.
5) Matrix $\boldsymbol{A}$ further constrained so that its bipartite graph has large girth.
6) Matrix $\boldsymbol{A} = [\boldsymbol{C}_1|\boldsymbol{C}_2]$ further constrained or slightly modified so that $\boldsymbol{C}_2$ is an invertible matrix.

Our proofs use the first ensemble. Our demonstrations use matrices from ensembles 4), 5), and 6).

The properties of the decoding problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \bmod 2$ also depend on the assumed noise model. We will give theoretical results for three cases. First, we give a general theorem for a broad class of symmetric noise models with memory (Definition 6). Second, we discuss a popular special case, the memoryless binary-symmetric channel, corresponding, in the case of Gallager codes, to a vector $\boldsymbol{x}$ of uniform density f. Third, the generalization to channels with continuous outputs is discussed in Appendix A.

B. Decodability for Arbitrary $P(\boldsymbol{x})$

To avoid confusion between Gallager and MN codes when discussing their common decoding problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \bmod 2$, we refer to the number of columns in $\boldsymbol{A}$ as L and the number of rows as M. (For a glossary of all symbols used in this paper, see Table III.) In the case of Gallager codes, $\boldsymbol{x}$ is a sample from the noise model, $\boldsymbol{x} = \boldsymbol{n}$. In the case of MN codes, $\boldsymbol{x}$ is the concatenation of the vectors $\boldsymbol{s}$ and $\boldsymbol{n}$, and the probability of $\boldsymbol{x}$ is separable into $P(\boldsymbol{x}) = P_{\mathrm{s}}(\boldsymbol{s})P_{\mathrm{n}}(\boldsymbol{n})$.

Let the ratio of the length of $\boldsymbol{x}$ to the length of $\boldsymbol{z}$ be $\lambda \equiv L/M$. The decoding problem is equivalent to playing a game in which a vector $\boldsymbol{x}$ is drawn from a probability distribution $P(\boldsymbol{x})$, and the vector $\boldsymbol{z} = \boldsymbol{Ax} \bmod 2$ is revealed; the player's task is then to identify the original vector $\boldsymbol{x}$, given the "encoding" $\boldsymbol{z}$, the matrix $\boldsymbol{A}$, and the known distribution P. The optimal decoder is an algorithm that identifies the vector $\boldsymbol{x}$ that maximizes $P(\boldsymbol{x})$ subject to the constraint $\boldsymbol{z} = \boldsymbol{Ax} \bmod 2$.

There is a Shannon limit for this game beyond which we cannot hope to recover $\boldsymbol{x}$ from $\boldsymbol{z}$ reliably. The maximum information content of $\boldsymbol{z}$ is clearly M shannons. We assume that the probability distribution of the noise is stationary and ergodic so that a mean entropy H_x can be defined for the distribution $P(\boldsymbol{x})$. Then the Shannon limit says reliable recovery of $\boldsymbol{x}$ from $\boldsymbol{z}$ is only possible if $LH_x \le M$, i.e.,

$$\lambda H_x \le 1. \tag{15}$$

If there are matrices $\boldsymbol{A}$ for which we can play the decoding game well at a value of λH_x close to this limit, then there exist Gallager codes which can communicate correspondingly close to the Shannon limit of a noisy channel whose noise distribution is given by $P(\boldsymbol{x})$.

For brevity we introduce the following definition.

Definition 7: A *satisfactory $(M, \lambda, t, \epsilon)$ matrix $\boldsymbol{A}$ for the distribution $P(\boldsymbol{x})$* is a matrix $\boldsymbol{A}$ having M rows and $L \ge \lambda M$ columns with weight t or less per column, with the following property: if $\boldsymbol{x}$ is generated from $P(\boldsymbol{x})$, the optimal decoder from $\boldsymbol{z} = \boldsymbol{Ax} \bmod 2$ back to $\hat{\boldsymbol{x}}$ achieves a probability of block error less than ϵ.

The following theorems will be proved.

Theorem 1—Good Codes: Given an integer $t \ge 3$ and a ratio $\lambda > 1$, there exists an entropy $H_x(\lambda, t) > 0$ such that, for any $P(\boldsymbol{x})$ of mean entropy $H_x < H_x(\lambda, t)$, and a desired block error probability $\epsilon > 0$, there exists an integer M and a satisfactory $(M, \lambda, t, \epsilon)$ matrix $\boldsymbol{A}$ for the distribution $P(\boldsymbol{x})$.

Theorem 2—Very Good Codes: Given a distribution $P(\boldsymbol{x})$ of mean entropy $H_x < 1$ and a desired $\lambda < 1/H_x$, there exists an integer $t(H_x, \lambda) \ge 3$ such that for any desired block error probability $\epsilon > 0$, there is an integer $M_{\min}$ such that for any $M > M_{\min}$, there is a satisfactory $(M, \lambda, t(H_x, \lambda), \epsilon)$ matrix $\boldsymbol{A}$ for the distribution $P(\boldsymbol{x})$.

Implications of Theorems 1 and 2 for Gallager codes. The first theorem effectively states that Gallager codes with any value of $t \ge 3$ are *good*, i.e., for any channel with appropriate entropy, there are Gallager codes which can achieve virtually error-free transmission at rates up to some nonzero rate $1 - 1/\lambda$, if the blocklength L is made sufficiently large.

The second theorem effectively states that Gallager 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/M arbitrarily small.

In Section II-C we prove these theorems, that is, we derive expressions for a function $H_x(\lambda, t) > 0$ satisfying Theorem 1, and a function $t(H_x, \lambda)$ satisfying Theorem 2. We also give numerical results relating to these theorems. Let the largest function for which the first theorem is true be $H_x^{\max}(\lambda, t)$. In Section II-D we evaluate a tighter numerical lower bound for $H_x^{\max}(\lambda, t)$. These are worst case results, true for *any* source of mean entropy H_x. In Section II-E we give numerical results for the case of the binary-symmetric channel, where considerably more optimistic bounds can be derived.

We also prove the following minimum distance theorem for Gallager codes which uses the function $H_x(\lambda, t)$ of Theorem 1.

Theorem 3—Good Distance Properties: Given an integer $t \ge 3$, a fraction $\delta < 0.5$, and a λ such that $H_2(\delta) < H_x(\lambda, t)$, there exist integers M and $L \ge \lambda M$ and a matrix $\boldsymbol{A}$ having M rows and L columns with weight t or less per column, such that the Gallager code with parity-check matrix $\boldsymbol{A}$ has minimum distance at least δL.

We can also prove that the Gilbert minimum-distance bound can be attained as $t \to \infty$, still with $\boldsymbol{A}$ very sparse.

Theorem 4—Gilbert Minimum Distance Bound Attainable: Given a fraction $\delta < 0.5$, and a λ such that $H_2(\delta) < 1/\lambda$, there exists a t and an $M_{\min}$ such that for any $M > M_{\min}$ there is a matrix $\boldsymbol{A}$ having M rows and $L \ge \lambda M$ columns with weight t or less per column, such that the Gallager code with parity-check matrix $\boldsymbol{A}$ has minimum distance at least δL.

Implication of Theorem 3 contrasted with Theorem 1. If one only aims to decode noise patterns of weight up to half of the minimum distance $d = \delta L$ (as is conventional in much of coding theory), then one can only handle noise levels up to $(d/L)/2$. But in fact the optimal decoder can decode (with vanishing probability of error) at noise levels up to (d/L). Thus Gallager codes can serve as good codes at noise levels *twice as great* as the maximum noise level that is attainable if one restricts attention to bounded distance decoding. The intuition for this result is that in a very-high-dimensional binary space, while two spheres of radius r whose centres are a distance d apart have a *nonzero* volume of intersection for any r greater than $d/2$, the *fractional* volume of intersection is *vanishingly small* as long as r is less than d.

Gallager codes, as Gallager showed [26] and we will show later, can in practice be decoded beyond their minimum distance.

C. Proof of Theorem 1

Consider the problem, given $\boldsymbol{A}$ and $\boldsymbol{z}$, of inferring $\boldsymbol{x}$, where $\boldsymbol{Ax} = \boldsymbol{z} \bmod 2$, and $\boldsymbol{x}$ has probability distribution $P(\boldsymbol{x})$ with mean entropy H_x. We consider the probability of error of a *typical set decoder* [16], averaging it over all very sparse random matrices $\boldsymbol{A}$. We establish the existence of a function $H_x(\lambda, t) > 0$ such that the probability of error can be bounded by a sum of terms which decrease as inverse powers of L, if $H_x < H_x(\lambda, t)$.

Typical set decoder. We consider the typical set

$$T = T_{(L,\eta)} = \left\{ \boldsymbol{x} \in \{0,1\}^L : \left| \frac{1}{L} \log_2 \frac{1}{P(\boldsymbol{x})} - H_x \right| \le \eta \right\} \tag{16}$$

where η is a small constant to be fixed later. Since

$$\sum_{\boldsymbol{x} \in T} P(\boldsymbol{x}) \le 1$$

the number of elements in this typical set, $|T|$, satisfies

$$|T| \leq 2^{L(H_x+\eta)}. \tag{17}$$

We now consider decoding $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} \bmod 2$ by the following procedure:

```
if there is a unique $\hat{\boldsymbol{x}} \in T$ such that $\boldsymbol{z} = \boldsymbol{A}\hat{\boldsymbol{x}} \bmod 2$ then
    produce $\hat{\boldsymbol{x}}$ as the decoding of $\boldsymbol{z}$;
else report decoding failure.
```

There are two failure modes for this decoder, with probabilities P_{I} and $P_{\mathrm{II}}(\boldsymbol{A})$ such that

$$P(\text{Block error} \,|\, \boldsymbol{A}) = P_{\mathrm{I}} + P_{\mathrm{II}}(\boldsymbol{A}). \tag{18}$$

I. Original vector not typical. This failure occurs with probability $P_{\mathrm{I}} = P(\boldsymbol{x} \notin T)$. Because $\boldsymbol{x}$ is assumed to have a mean entropy H_x, this probability vanishes as $L \to \infty$ (see Definition 5, Section I-B).

II. $\boldsymbol{x}$ is typical but at least one other typical vector $\boldsymbol{x}'$ has the same encoding $\boldsymbol{z}$. We now concentrate on the probability of a decoding error arising this way, and denote the average of $P_{\mathrm{II}}(\boldsymbol{A})$ over $\boldsymbol{A}$ drawn from ensemble 1) (Section II-A) by $\bar{P}_{\mathrm{II}}$.

We define the indicator function $\delta(S)$ to have the value 1 if the proposition S is true and 0 otherwise. We can bound $P_{\mathrm{II}}(\boldsymbol{A})$ by

$$P_{\mathrm{II}}(\boldsymbol{A}) \leq \sum_{\boldsymbol{x} \in T} P(\boldsymbol{x}) \sum_{\substack{\boldsymbol{x}' \in T \\ \boldsymbol{x}' \neq \boldsymbol{x}}} \delta[\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}') = 0 \bmod 2]. \tag{19}$$

In (19) the second sum is the number of typical vectors $\boldsymbol{x}'$ that have the same encoding as $\boldsymbol{x}$. We now average over codes $\boldsymbol{A}$.

$$\bar{P}_{\mathrm{II}} \leq \sum_{\substack{\boldsymbol{x}, \boldsymbol{x}' \in T \\ \boldsymbol{x}' \neq \boldsymbol{x}}} P(\boldsymbol{x}) \left\{ \sum_{\boldsymbol{A}} P(\boldsymbol{A}) \delta[\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}') = 0] \right\}. \tag{20}$$

The term in brackets only depends on the weight w of the difference $(\boldsymbol{x} - \boldsymbol{x}')$. The probability that $\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}') = 0$ is the probability that w columns of the very sparse matrix $\boldsymbol{A}$ sum to zero. Because $\boldsymbol{A}$ is constructed by flipping t bits per column at random with replacement, this is the probability that wt steps of the random walk on the M-dimensional hypercube, starting from the origin, bring us back to origin. We denote this probability $p_{00}^{(wt)}$ and define $h(w \,|\, \boldsymbol{x})$ to be the number of typical vectors $\boldsymbol{x}'$ such that the difference $\boldsymbol{x} - \boldsymbol{x}'$ has weight w, for the given $\boldsymbol{x}$. Then we have

$$\bar{P}_{\mathrm{II}} \leq \sum_{w=1}^{L} \sum_{\boldsymbol{x} \in T} P(\boldsymbol{x}) h(w \,|\, \boldsymbol{x}) p_{00}^{(wt)}. \tag{21}$$

In Appendix E we give an expression for the function $p_{00}^{(r)}$; this function is zero for all odd r, and is a decreasing function of even r. It will be convenient to introduce an upper bound on $p_{00}^{(r)}$ which is equal to it for even r and which is a decreasing function. We define this function, $q_{00}^{(r)}$, in (87). In Appendix E we also derive various upper bounds on $p_{00}^{(r)}$ and $q_{00}^{(r)}$ from which we will use the following two:

$$p_{00}^{(r)} \leq q_{00}^{(r)} \leq q_d^{(r)}$$

$$\equiv \begin{cases} \left(\frac{3}{4}\frac{r}{M}\right)^{r/2}, & \text{for } r \leq r' = \gamma' M, \text{ where } \gamma' = \frac{4}{3e} \\ \left(\frac{3}{4}\gamma'\right)^{r'/2}, & \text{for } r > r'; \end{cases} \tag{22}$$

$$\log q_{00}^{(r)} \leq \log q_b^{(r)} \equiv M(e^{-2r/M} - \log 2) + \log 2. \tag{23}$$

These bounds are tightest for $r \ll M$ and $r \gg M$, respectively. Both these bounds are decreasing functions of r.

We now have

$$\bar{P}_{\mathrm{II}} \leq \sum_{w=1}^{L} \sum_{\boldsymbol{x} \in T} P(\boldsymbol{x}) h(w \,|\, \boldsymbol{x}) q_{00}^{(wt)}. \tag{24}$$

We pause to dissect the product

$$\left[\sum_{\boldsymbol{x} \in T} P(\boldsymbol{x}) h(w \,|\, \boldsymbol{x}) \right] \left[q_{00}^{(wt)} \right].$$

The first factor is typically a rapidly increasing function of w up to some peak (which in the case of a binary-symmetric channel of density f is located at $w \simeq 2f(1-f)L$). The second factor $q_{00}^{(wt)}$ is largest for $w = 1$ and falls initially as a power law $1/M^{wt/2}$ (see (22)) decreasing to an equilibrium value of 2×2^{-M} (see (23)), corresponding to an equilibrium distribution uniform over (half of) the states of the hypercube. We want the product of these two factors to be vanishingly small (in increasing L) for all w.

Intuitively, if the product were large at small w, then type II errors would arise because there would be a small subset of columns in $\boldsymbol{A}$ that sum to zero such that it is possible to confuse vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ that differ in only a few bits (and which could therefore both be typical). If the product is large at large w, then type II errors would arise because there are two completely different vectors $\boldsymbol{x}, \boldsymbol{x}'$ which have the same encoding. Gallager codes and MN codes are good because we can indeed make the product vanishingly small for all w.

1) Worst Case Analysis: Up to this point we have not mentioned any form for $P(\boldsymbol{x})$, which determines the function $h(w \,|\, \boldsymbol{x})$ in (24). We now proceed with a worst case assumption for the function $h(w \mid \boldsymbol{x})$.

We know that

$$\sum_{w=1}^{L} h(w \,|\, \boldsymbol{x}) \leq |T| \leq 2^{L(H_x+\eta)}.$$

Also $h(w \,|\, \boldsymbol{x})$ is bounded above by the maximum number of distinct vectors of weight w, $\binom{L}{w}$. Finally, $q_{00}^{(wt)}$ is a decreasing function of w. So we can replace the function $h(w \,|\, \boldsymbol{x})$ in (24) by the worst case function $h^*(w)$ which has the maximum value at small w

$$h^*(w) = \begin{cases} \binom{L}{w}, & w \leq w^* \\ 0, & w > w^* \end{cases} \tag{25}$$

where w^* is chosen to satisfy $\sum_{w=1}^{w^*} h^*(w) \geq 2^{L(H_x+\eta)}$. If we write

$$H_2(w^*/L) = H_x + \eta' \tag{26}$$

where η' is a small quantity satisfying $\eta' > \eta$ and $\eta' < 1 - H_x$, then

$$\sum_{w=1}^{w^*} h^*(w) \geq \binom{L}{w^*} \geq \frac{1}{L+1} 2^{LH_2(w^*/L)} \geq 2^{L(H_x+\eta)} \tag{27}$$

as long as η is sufficiently small and L is sufficiently large that $\eta + \frac{1}{L}\log_2(L+1) < \eta'$. (Here we have used (73) from Appendix D.) Notice that the only dependence that the function $h^*(w)$ has on the source distribution $P(\boldsymbol{x})$ is the cutoff value w^*, which is controlled by H_x. Thus

$$\bar{P}_{\rm II} \leq \sum_{\boldsymbol{x} \in T} P(\boldsymbol{x}) \sum_{w=1}^{w^*} \binom{L}{w} q_{00}^{(wt)} \tag{28}$$

$$\leq \sum_{w=1}^{w^*} \binom{L}{w} q_{00}^{(wt)}. \tag{29}$$

We now seek conditions under which $\bar{P}_{\rm II}$ can be made arbitrarily small by increasing M. Here first is a sketch of our proof. Consider the logarithm of the term inside the summation in (29) as a function of w. Using (73) again

$$\log\left[\binom{L}{w} q_{00}^{(wt)}\right] \leq M\left(\lambda H_2^e(w/L) + \frac{1}{M}\log q_{00}^{(wt)}\right). \tag{30}$$

We want to show that for all w less than some $w^{\max}$ that grows linearly with M, $w^{\max} = \beta M$, the above term becomes increasingly large and negative as M increases. We need all the terms in the sum in (29) to go to zero, and we need almost all of them to do so faster than $1/M$, so that the sum itself goes to zero. To aid the intuition, the two terms $\lambda H_2^e(w/L)$ and $\frac{1}{M}\log q_e^{(wt)}$ (a tight numerical upper bound on $\log q_{00}^{(wt)}$) are plotted in Fig. 1 as a function of w/L for a variety of values of t, for $\lambda = 2$. The function $\lambda H_2^e(w/L)$ is an increasing function of w (if $w < L/2$) reaching a maximum value of $\lambda \log 2$ at $w = L/2$; the function $\frac{1}{M}\log q_{00}^{(wt)}$ is a decreasing function of w reaching a minimum value of $-\log 2$ asymptotically. For $w/L = 1/2$, the right-hand side of (30) is certainly positive since $\lambda > 1$. For small w/L, on the other hand (starting at $w = 1$), the sum is negative, if $t > 2$, as we now prove. There therefore exists a $\beta > 0$ such that, for all $w < \beta M$, the sum is negative. This defines the critical value of w^* which, via (26), proves the existence of the function $H_x(\lambda, t)$ that we seek.

We continue from (29), using (22) and inequality (85) and assuming $w^* t < \frac{4}{3e}M$.

$$\bar{P}_{II} \leq \sum_{w=1}^{w^*} \binom{L}{w} q_{00}^{(wt)} \leq \sum_{w=1}^{w^*} \frac{L^w}{w!}\left(\frac{3}{4}\frac{wt}{M}\right)^{wt/2} \tag{31}$$

$$\leq \sum_{w=1}^{w^*} M^w \frac{\lambda^w}{w^w e^{-w} e}\left(\frac{3}{4}\frac{wt}{M}\right)^{wt/2} \tag{32}$$

$$\leq \sum_{w=1}^{w^*} \left(\frac{w}{M}\right)^{\frac{wt}{2}-w} (e\lambda)^w \left(\frac{3}{4}t\right)^{wt/2} \tag{33}$$

$$\leq \sum_{w=1}^{w^*} \left(c\frac{w}{M}\right)^{aw} \tag{34}$$

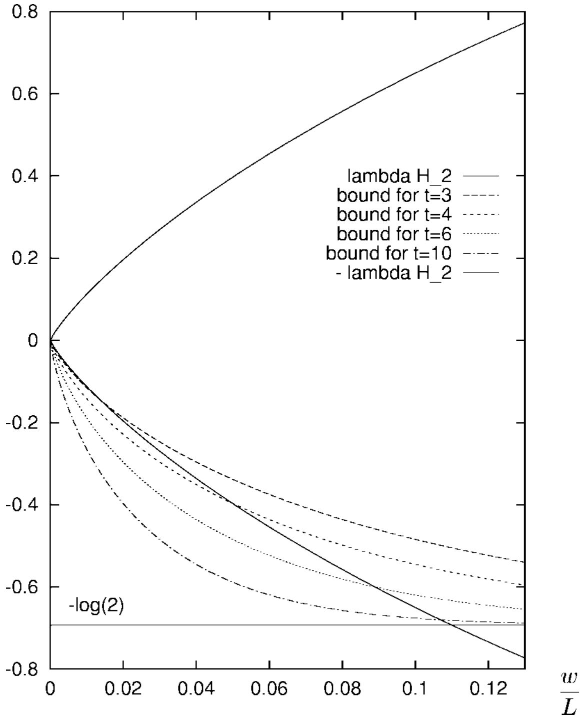


Fig. 1. Plot of a numerical upper bound $\frac{1}{M}\log q_e^{(wt)}$ on the function $\frac{1}{M}\log q_{00}^{(wt)}$, for large M and for various values of t; and the function $\lambda H_2^e(w/L)$, as a function of w/L, for $\lambda = 2$. For convenience, the function $-\lambda H_2^e(w/L)$ is also shown. The intersection point of the two curves $-\lambda H_2^e(w/L)$ and $\frac{1}{M}\log q_e^{(wt)}$ defines a value of w^* which gives a lower bound on the achievable information rate for Gallager and MN codes. This graph gives an informal proof of Theorems 1 and 2 for the case $\lambda = 2$. The content of Theorem 1 is that for any t the solid line $-\lambda H_2^e(w/L)$ lies above the dotted line $\frac{1}{M}\log q_e^{(wt)}$ for all w/L up to some nonzero value. The content of Theorem 2 is that as $t \to \infty$, the first point of intersection of the two curves approaches the point at which $\lambda H_2^e(w/L) = \log 2$.

where $a = (\frac{t}{2} - 1)$ satisfies $a > 0$ if $t > 2$, and

$$c = \left[e\lambda\left(\frac{3t}{4}\right)^{t/2}\right]^{1/a}.$$

Now as a function of w, $\log(c\frac{w}{M})^{aw}$ is a decreasing function for $w \leq \frac{M}{ce}$, as is easily confirmed by differentiation, so if we constrain w^* by $w^* \leq \frac{M}{ce}$, we can bound the last $(w^* - 2)$ terms by $\frac{M}{ce}$ times the third term

$$\bar{P}_{\rm II} \leq \sum_{w=1}^{w^*} \binom{L}{w} q_{00}^{(wt)}$$

$$\leq \left(\frac{c}{M}\right)^{(\frac{t}{2}-1)} + \left(\frac{2c}{M}\right)^{2(\frac{t}{2}-1)} + \frac{M}{ce}\left(\frac{3c}{M}\right)^{3(\frac{t}{2}-1)}. \tag{35}$$

Thus the average probability of error for this ensemble of codes vanishes with increasing M as $M^{-1/2}$ if $t = 3$ and faster if $t \geq 3$. There therefore exists a β given by $\beta = \min(\frac{4}{3et}, \frac{1}{ce}) > 0$ such that if $w^* < \beta M$, or equivalently, if $H_x + \eta' < H_2(\beta/\lambda)$, then

$$\bar{P}_{\rm II} \leq \sum_{w=1}^{w^*} \binom{L}{w} q_{00}^{(wt)} \to 0 \quad \text{as } M \to \infty. \tag{36}$$

This establishes the existence of the function $H_x(\lambda, t)$, since if we define

$$H_x(\lambda, t) = H_2(\beta/\lambda) \tag{37}$$

then, for any $H_x < H_x(\lambda, t)$, we can set η', η, and L such that $\eta' > \eta + \frac{1}{L}\log_2(L+1)$ and $\eta' < H_x(\lambda, t) - H_x$.

This completes the proof of Theorem 1. We note that the value of $H_x(\lambda, t)$ given above is not the largest such function $H_x^{\max}(\lambda, t)$ that satisfies the theorem. By using tighter bounds on $p_{00}^{(wt)}$ we can prove the theorem for larger values of $H_x(\lambda, t)$. We use a numerical bound to obtain graphs of lower bounds on $H_x^{\max}(\lambda, t)$ in Section II-D.

Comment. It is instructive to study the terms which dominate the sum over w. The terms with small w decrease as inverse powers of M as long as $(w - wt/2) < 0$ for all $w \geq 1$, i.e., as long as $t > 2$. The $w = 1$ term, the largest, decreases as $M^{1-t/2}$. We can get a stronger decrease of $\bar{P}_{\rm II}$ with M if we assume that each column has exactly t ones in it. Then we can omit the $w = 1$ term, which corresponds to the probability of there being a zero column in $\boldsymbol{A}$. By the further step of ensuring that no two columns of the matrix $\boldsymbol{A}$ are identical, we can rule out the $w = 2$ term from the sum. By adding further constraints when generating the code, that the girth of the corresponding bipartite graph must be large, we can make the probability of this sort of "small w" error even smaller. This is the motivation for the code Constructions 1B and 2B that we introduce later on.

2) Proof of Theorem 3: A code with parity-check matrix $\boldsymbol{H}$ has minimum distance d if and only if any $(d-1)$ columns of $\boldsymbol{H}$ are linearly independent, and there exist d linearly dependent columns. We can prove the existence of a matrix $\boldsymbol{A}$ with δL linearly independent columns easily starting from (36). Here δL is the desired minimum distance. We set $w^* = \delta L$ and note that the assumption of the theorem, $H_2(\delta) < H_x(\lambda, t)$ implies that the quantity $\sum_{w=1}^{w^*} \binom{L}{w} p_{00}^{(wt)}$ vanishes with increasing L.

The quantity $\sum_{w=1}^{w^*} \binom{L}{w} p_{00}^{(wt)}$ is the expected number of linear dependences of up to w^* columns in a randomly generated matrix $\boldsymbol{A}$. Since this expected number is less than 1 (indeed, it goes to zero), there must be at least one $\boldsymbol{A}$ in the ensemble such that the number of linear dependences is zero (indeed, nearly all $\boldsymbol{A}$ in the ensemble must satisfy this).

3) Proof of Theorem 2: We now show that if we are free to choose t then Gallager and MN codes can get arbitrarily close to the Shannon limit. We prove that for a desired λ and any source with mean entropy H_x such that $\lambda H_x < 1$, and any $\epsilon > 0$, there exists a t and an $M_{\min}$ such that for any $M > M_{\min}$, the corresponding decoding problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \bmod 2$ can be solved with probability of failure less than ϵ, averaged over the first ensemble of matrices $\boldsymbol{A}$.

We start from (29), which bounds the type II probability of error thus

$$\bar{P}_{\rm II} \leq \sum_{w=1}^{w^*} \binom{L}{w} q_{00}^{(wt)} \tag{38}$$

where w^* is given by

$$H_2(w^*/L) = H_x + \eta' \tag{39}$$

and $\eta' \geq \eta + \frac{1}{L}\log_2(L+1)$. From (30),

$$\bar{P}_{\rm II} \leq \sum_{w=1}^{w^*} \exp\left[M\left(\lambda H_2^e(w/L) + \frac{1}{M}\log q_{00}^{(wt)}\right)\right]. \tag{40}$$

If we can set t and η' such that the term in parentheses is negative for all $w \leq w^*$ then $\bar{P}_{\rm II}$ goes to zero as M increases. For large t it is evident (c.f. Fig. 1) that $(\lambda H_2^e(w/L) + \frac{1}{M}\log q_{00}^{(wt)})$ attains its largest value at $w = w^*$. So, substituting upper bound (23) for $\log q_{00}^{(r)}$, our condition for vanishing $\bar{P}_{\rm II}$ is

$$\lambda H_2^e(w^*/L) + (e^{-2w^*t/M} - \log 2) + O(1/M) \leq 0. \tag{41}$$

Substituting in (39), we require

$$e^{-2H_2^{-1}(H_x+\eta')\lambda t} + O(1/M) \leq \log 2 - \lambda(H_x + \eta')\log 2 \tag{42}$$

so that if we set η' such that $\lambda(H_x + \eta') < 1$ and t such that

$$\exp[-2H_2^{-1}(H_x + \eta')\lambda t] < \log 2[1 - \lambda(H_x + \eta')]$$

then $\bar{P}_{\rm II}$ vanishes with increasing M, so an $M_{\min}$ can be found such that the average error probability of the ensemble of matrices $\boldsymbol{A}$ is below any desired error probability ϵ.

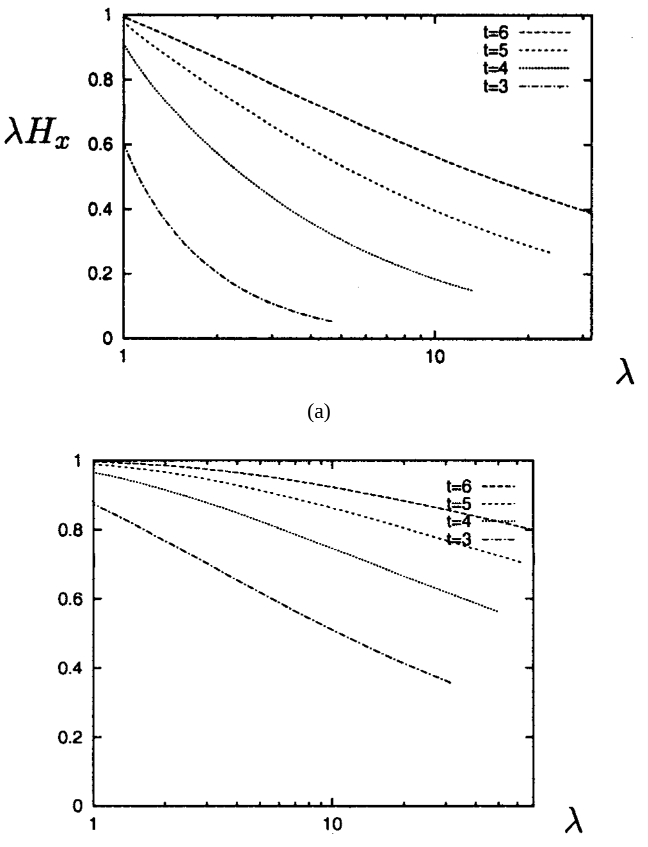
4) Proof of Theorem 4: This theorem follows from Theorem 2 as Theorem 3 followed from Theorem 1. Briefly, since we can find a t such that $\sum_{w=1}^{w^*} \binom{L}{w} p_{00}^{(wt)}$ vanishes with increasing L for any w^*/L satisfying $H_2(w^*/L) < 1/\lambda$, this implies that, for sufficiently large L, we can find a matrix with at least δL linearly independent columns for any δ satisfying $H_2(\delta) < 1/\lambda$.

D. Numerical Bounds on $H_x^{\max}(\lambda, t)$: Worst Case

Theorem 1 proved the existence, for a given ratio λ, of an entropy $H_x(\lambda, t)$ such that there exist good Gallager and MN codes with rates defined by λ for *any* channel with symmetric stationary ergodic noise (Definition 6) with mean entropy up to $H_x(\lambda, t)$. We define $H_x^{\max}(\lambda, t)$ to be the largest function for which the theorem holds. In Fig. 2(a) we show a numerically obtained bound on this function. The graph shows a lower bound on $\lambda H_x^{\max}(\lambda, t)$ as a function of λ; the Shannon limit provides the upper bound $\lambda H_x^{\max}(\lambda, t) \leq 1$. This graph was obtained as follows. Using the expansion of $p_{00}^{(r)}$ in (86), the condition for $\bar{P}_{\rm II}$ to vanish is that for all $w < w^*$ and all j the quantity

$$G = \log\left[\binom{L}{w}\binom{M}{j}\left(1 - \frac{2j}{M}\right)^{wt}\right] \tag{43}$$

should be negative. We use the inequality $\binom{N}{K} \leq 2^{NH_2(K/N)}$ and maximize G as a function of $\phi = w/L$ and $\kappa = j/M$. We repeat this procedure iteratively to locate the critical value of w^*/L such that G is zero. This gives the bound.



(a)

(b)

Fig. 2. Lower bounds on $\lambda H_x^{\max}(\lambda, t)$ (a) for arbitrary source $P(\boldsymbol{x})$ and (b) assuming $P(\boldsymbol{x})$ corresponding to binary-symmetric channel, for matrices $\boldsymbol{A}$ with t from 3 to 6. As the weight per column t increases, the achievable rates rise towards the Shannon limit $\lambda H_x = 1$.

E. Numerical Upper and Lower Bounds: Uniform Noise Model

We now assume that the channel under discussion is the binary-symmetric channel and obtain numerical upper and lower bounds on the achievable rate of Gallager codes, assuming the optimal decoder. The method here is as in the previous section, with the worst case function $h^*(w) = \binom{L}{w}$ replaced by the $h(w)$ appropriate to the uniform noise case, as detailed in Appendix F. The resulting lower bounds on $H_x^{\max}(\lambda, t)$ are shown in Fig. 2(b).

1) Achievable Rate for Gallager Codes over Binary-Symmetric Channel: From the lower bound on the noise entropy plotted in Fig. 2(b) we can compute a lower bound on the achievable communication rate using Gallager codes (given an optimal decoder), as a function of the noise level of the binary-symmetric channel. This rate is shown in shannons in Fig. 3 for $t = 3, 4, 5, 6$ and compared with the capacity. As the weight per column t increases the bounds rise towards the capacity.

2) Upper Bounds on the Communication Rate over a Binary-Symmetric Channel Fig. 4(a) and (b) addresses the question "what information rate is definitely *not* achievable for a given t and noise level f?" This is a relatively easy question to answer, and it gives insight into what is lost in the above proof by switching from the constrained ensemble of random matrices $\boldsymbol{A}$ with weight t per column and $t_r \equiv \lambda t$ per row to the unconstrained ensemble of matrices. It also gives insight into why Gallager codes with fixed t are only good codes and not very good codes.

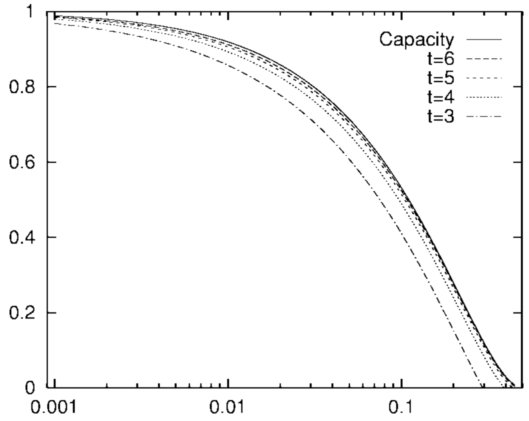
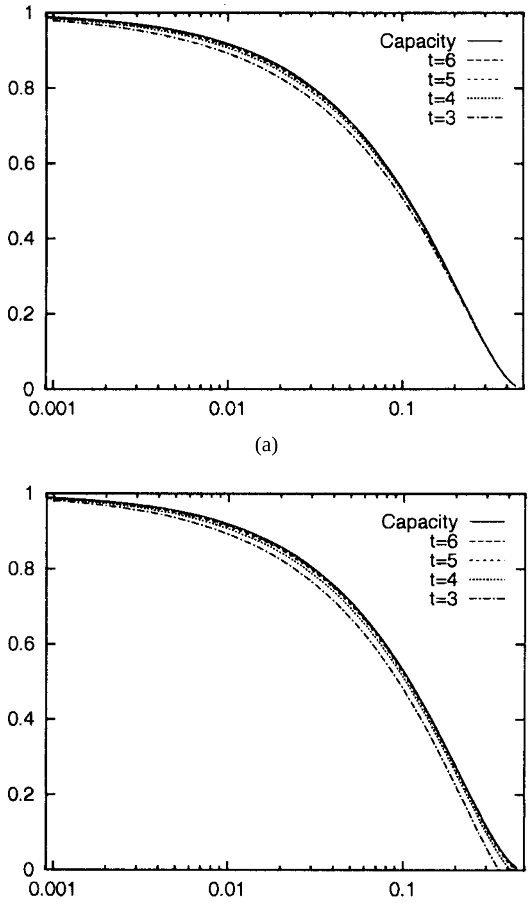


Fig. 3. Lower bounds on achievable information rate (in shannons) versus binary-symmetric channel's noise level f for Gallager codes (ensemble 1) with t from 3 to 6. The solid line shows the channel capacity.



(a)

(b)

Fig. 4. Upper bound on the achievable information rate in shannons versus binary-symmetric channel's noise level, for Gallager codes, compared with the channel capacity. (a) This bound was obtained assuming that matrices $\boldsymbol{A}$ were drawn from the ensemble having uniform weight per row as well as per column (Ensemble 3). (b) This bound was obtained assuming that matrices $\boldsymbol{A}$ were drawn from the ensemble having uniform weight per column only (Ensemble 2). The graphs differ significantly at high noise levels.

Consider the decoding problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \bmod 2$, where $\boldsymbol{x}$ has density f and length λM, and $\boldsymbol{z}$ has length M. It is clearly impossible to solve the problem of deducing $\boldsymbol{x}$ reliably from $\boldsymbol{z}$

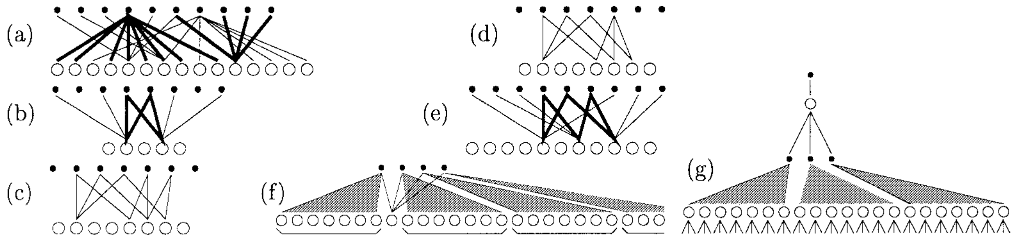


Fig. 5. The vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ viewed as nodes in a belief network. White circles denote bits x_l. Black dots denote checks z_m. We illustrate the case $t = 4$, $t_r = 8$. (a) This figure emphasizes with bold lines the four connections from one bit and the eight connections to one check. Every bit x_l is the parent of four checks z_m, and each check z_m is the child of eight bits. (b)–(e) Certain topological structures are undesirable in the network defined by the matrix $\boldsymbol{A}$: in (b) there is a cycle of length 4 in the network; we can forbid this topology by saying that the overlap between two columns of $\boldsymbol{A}$ must not exceed 1; in (c), (d), and (e) more complex topologies are illustrated. Many of our experiments have used matrices $\boldsymbol{A}$ 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 special cases. This means that every bit's "coparents" (other bits that are parents of its children) consist of t nonoverlapping sets of bits as shown in (f). (g) A fragment of an infinite belief network with $t = 4$ and $t_r = 8$ and no cycles. In Section III-C2 we analyze the flow of information up this network.

if the information content of $\boldsymbol{x}$, $\lambda M H_2(f)$ shannons, is more than the information content of $\boldsymbol{z}$. We get the Shannon bound by noting that $H(\boldsymbol{z})$, in shannons, is less than or equal to its length M. But we can get upper bounds lower than this by tightening the bound on $H(\boldsymbol{z})$.

Probability distribution of the random variable z_m. We refer to the elements z_m corresponding to each row $m = 1 \cdots M$ of $\boldsymbol{A}$ as checks and to the elements of $\boldsymbol{x}$ as bits. Let z_m be the sum of τ bits of density f. We define $p_z(\tau)$ to be the probability that $z_m = 1$. Starting from $p_z(0) = 0$, we can use the recurrence relation

$$p_z(\tau + 1) = p_z(\tau)(1 - f) + (1 - p_z(\tau))f \qquad (44)$$

to obtain [26]

$$p_z(\tau) = \frac{1}{2} - \frac{1}{2}(1 - 2f)^{\tau}. \qquad (45)$$

We use this result to obtain tighter bounds on the achievable communication rate.

Bound for constrained matrices $\boldsymbol{A}$. Consider the ensemble of random matrices $\boldsymbol{A}$ with weight t per column and as near as possible t_r per row (Ensemble 3 in Section II-A). In the general case where $t_r = \lambda t$ is not an integer, the information content per check of $\boldsymbol{z}$ is bounded above by the average of the marginal information content of one check, averaged over the ensemble, that is,

$$(t_r - \lfloor t_r \rfloor) H_2(p_z(\lfloor t_r \rfloor + 1)) + (1 - t_r + \lfloor t_r \rfloor) H_2(p_z(\lfloor t_r \rfloor)). \qquad (46)$$

This gives the bound shown in Fig. 4(a), which was created as follows. For a selected value of t and f, a search was made over λ for a value such that the upper bound on the information content per check of $\boldsymbol{z}$ is just above $\lambda H_2(f)$, using (46). The graph is then a plot of $(1 - 1/\lambda)$ versus f.

Bound for unconstrained matrices $\boldsymbol{A}$. Now what if we remove the constraint λt per row, reverting to Ensemble 2 of Section II-A? Intuitively we can see that $H(\boldsymbol{z})$ will decrease. Some checks will be sums of more than $\lfloor t_r \rfloor + 1$ bits of $\boldsymbol{x}$, and some will be sums of fewer than $\lfloor t_r \rfloor$. The former checks will have a value of p_z slightly closer to 0.5, whereas the latter will have values of p_z further from 0.5. Some checks may be the sum of no bits at all (with probability $\exp(-t_r)$), so that they convey no information. The convexity of the relevant functions produces a net decrease in the information content of $\boldsymbol{z}$. To work out a bound for the unconstrained ensemble, we sum over all possible weights of a row τ, evaluate $H_2(p_z(\tau))$, and weight by the probability of τ, which is a Poisson distribution with mean t_r. The resulting upper bound is shown in Fig. 4(b). We see that the bound is lower than that for the constrained matrices, and looks similar to the lower bound in Fig. 3. It thus seems plausible that, were we to change from Ensemble 2 to Ensemble 3 in the main proof of the paper, we would be able to prove the achievability of somewhat larger rates for any given t.

This concludes our discussion of what would be achievable given an optimal decoder. We now move on to practical decoding algorithms.

III. Practical Decoding by the Sum-Product Algorithm

We have developed a "sum-product decoder," also known as a "belief propagation decoder" [51], [33] for the decoding problem $\boldsymbol{Ax} = \boldsymbol{z} \bmod 2$. In this work, we have rediscovered a method of Gallager [27]. See [68], [24], and [46] for further discussion of the sum-product algorithm.

We refer to the elements z_m corresponding to each row $m = 1 \cdots M$ of $\boldsymbol{A}$ as checks. We think of the set of bits $\boldsymbol{x}$ and checks $\boldsymbol{z}$ as making up a "belief network," also known as a "Bayesian network," "causal network," or "influence diagram" [51], in which every bit x_l is the parent of t checks z_m, and each check z_m is the child of t_r bits (Fig. 5). The network of checks and bits form a bipartite graph: bits only connect to checks, and *vice versa*.

We aim, given the observed checks, to compute the marginal posterior probabilities $P(x_l = 1 \,|\, \boldsymbol{z}, \boldsymbol{A})$ for each l. Algorithms for the computation of such marginal probabilities in belief networks are found in [51]. These computations are expected to be intractable for the belief network corresponding to our problem $\boldsymbol{Ax} = \boldsymbol{z} \bmod 2$ 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. This approach of ignoring cycles has been used in the artificial intelligence literature [4] but is now frowned upon because it produces inaccurate probabilities (D. Spiegelhalter, personal communication). However, for our problem the end product is a decoding; the marginal probabilities are not required if the decoding is correct. Also, the posterior probability, in the case of a good code communicating at an achievable rate, is expected typically to be hugely concentrated on the most probable decoding. And as the size $M \times L$ of the code's matrix $\boldsymbol{A}$ is increased, it becomes increasingly easy to produce matrices in which there are no cycles of any given length, so we expect that, asymptotically, this algorithm will be a good algorithm. We have obtained excellent results with M equal to 1000 and 10000. The algorithm often gives useful results after a number of iterations much greater than the number at which it could be affected by the presence of cycles.

A. The Algorithm

We have implemented the following algorithm (for background reading see [51]). The algorithm is appropriate for a binary channel model in which the noise bits are independent—for example, the memoryless binary-symmetric channel, or the Gaussian channel with binary inputs and real outputs (the connection to real-output channels is explained in Appendix A).

We denote the set of bits l that participate in check m by $\mathcal{L}(m) \equiv \{l : A_{ml} = 1\}$. Similarly, we define the set of checks in which bit l participates, $\mathcal{M}(l) \equiv \{m : A_{ml} = 1\}$. We denote a set $\mathcal{L}(m)$ with bit l excluded by $\mathcal{L}(m) \setminus l$. The algorithm has two alternating parts, in which quantities q_{ml} and r_{ml} associated with each nonzero element in the $\boldsymbol{A}$ matrix are iteratively updated. The quantity q_{ml}^{x} is meant to be the probability that bit l of $\boldsymbol{x}$ has the value x, given the information obtained via checks other than check m. The quantity r_{ml}^{x} is meant to be the probability of check m being satisfied if bit l of $\boldsymbol{x}$ is considered fixed at x and the other bits have a separable distribution given by the probabilities $\{q_{ml'} : l' \in \mathcal{L}(m) \setminus l\}$. The algorithm would produce the exact posterior probabilities of all the bits after a fixed number of iterations if the bipartite graph defined by the matrix $\boldsymbol{A}$ contained no cycles [51].

Initialization. Let $p_l^0 = P(x_l = 0)$ (the prior probability that bit x_l is 0), and let $p_l^1 = P(x_l = 1) = 1 - p_l^0$. In the case of a Gallager code and a binary-symmetric channel, p_l^1 will equal $f_{\rm n}$. In the case of an MN code, p_l^1 will be either $f_{\rm s}$ or $f_{\rm n}$, depending on whether bit l is part of the message or the noise. If the noise level varies in a known way (for example, if the channel is a binary input Gaussian channel with a real output) then p_l^1 is initialized to the appropriate normalized likelihood. For every (l, m) such that $A_{ml} = 1$ the variables q_{ml}^0 and q_{ml}^1 are initialized to the values p_l^0 and p_l^1, respectively.

Horizontal step. In the horizontal step of the algorithm, we run through the checks m and compute for each $l \in \mathcal{L}(m)$ two probabilities: first, r_{ml}^0, the probability of the observed value of z_m arising when $x_l = 0$, given that the other bits $\{x_{l'} : l' \neq l\}$ have a separable distribution given by the probabilities $\{q_{ml'}^0, q_{ml'}^1\}$, defined by:

$$r_{ml}^0 = \sum_{\{x_{l'}: l' \in \mathcal{L}(m) \setminus l\}} P(z_m \,|\, x_l = 0, \{x_{l'} : l' \in \mathcal{L}(m) \setminus l\}) \times \prod_{l' \in \mathcal{L}(m) \setminus l} q_{ml'}^{x_{l'}} \tag{47}$$

and second, r_{ml}^1, the probability of the observed value of z_m arising when $x_l = 1$, defined by

$$r_{ml}^1 = \sum_{\{x_{l'}: l' \in \mathcal{L}(m) \setminus l\}} P(z_m \,|\, x_l = 1, \{x_{l'} : l' \in \mathcal{L}(m) \setminus l\}) \times \prod_{l' \in \mathcal{L}(m) \setminus l} q_{ml'}^{x_{l'}}. \tag{48}$$

The conditional probabilities in these summations are either zero or one, depending on whether the observed z_m matches the hypothesized values for x_l and the $\{x_{l'}\}$.

These probabilities can be computed in various obvious ways based on (47) and (48). The computations may be done most efficiently (if $|\mathcal{L}(m)|$ is large) by regarding $z_m + x_l$ as the final state of a Markov chain with states 0 and 1, this chain being started in state 0, and undergoing transitions corresponding to additions of the various $x_{l'}$, with transition probabilities given by the corresponding $q_{ml'}^0$ and $q_{ml'}^1$. The probabilities for z_m having its observed value given either $x_l = 0$ or $x_l = 1$ can then be found efficiently by use of the forward–backward algorithm [7], [65], [5].

A particularly convenient implementation of this method uses forward and backward passes in which products of the differences $\delta q_{ml} \equiv q_{ml}^0 - q_{ml}^1$ are computed. We obtain $\delta r_{ml} \equiv r_{ml}^0 - r_{ml}^1$ from the identity

$$\delta r_{ml} = (-1)^{z_m} \prod_{l' \in \mathcal{L}(m) \setminus l} \delta q_{ml'}. \tag{49}$$

This identity is derived by iterating the following observation: if $\zeta = x_\mu + x_\nu \bmod 2$, and x_μ and x_ν have probabilities q_μ^0, q_ν^0 and q_μ^1, q_ν^1 of being 0 and 1, then $P(\zeta = 1) = q_\mu^1 q_\nu^0 + q_\mu^0 q_\nu^1$ and $P(\zeta = 0) = q_\mu^0 q_\nu^0 + q_\mu^1 q_\nu^1$. Thus

$$P(\zeta = 0) - P(\zeta = 1) = (q_\mu^0 - q_\mu^1)(q_\nu^0 - q_\nu^1).$$

Finally, note that $r_{ml}^0 + r_{ml}^1 = 1$, and hence $r_{ml}^0 = (1 + \delta r_{ml})/2$ and $r_{ml}^1 = (1 - \delta r_{ml})/2$.

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

$$q_{ml}^0 = \alpha_{ml} p_l^0 \prod_{m' \in \mathcal{M}(l) \setminus m} r_{m'l}^0 \tag{50}$$

$$q_{ml}^1 = \alpha_{ml} p_l^1 \prod_{m' \in \mathcal{M}(l) \setminus m} r_{m'l}^1 \tag{51}$$

where α_{ml} is chosen such that $q_{ml}^0 + q_{ml}^1 = 1$. These products can be efficiently computed in a downward pass and an upward pass.

We can also compute the "pseudoposterior probabilities" q_l^0 and q_l^1 at this iteration, given by

$$q_l^0 = \alpha_l p_l^0 \prod_{m \in \mathcal{M}(l)} r_{ml}^0 \tag{52}$$

$$q_l^1 = \alpha_l p_l^1 \prod_{m \in \mathcal{M}(l)} r_{ml}^1. \tag{53}$$

These quantities are used to create a tentative decoding $\hat{\boldsymbol{x}}$, the consistency of which is used to decide whether the decoding algorithm can halt.

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

Decoding. If the belief network really were a tree without cycles, the values of the pseudoposterior probabilities q_l^0 and q_l^1 at each iteration would correspond exactly to the posterior probabilities of bit l given the states of all the checks in a truncated belief network centred on bit l and extending out to a radius equal to twice the number of iterations. Our decoding procedure is to set $\hat{x}_l$ to 1 if $q_l^1 > 0.5$ and see if the checks $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{z} \bmod 2$ are all satisfied, halting when they are, and declaring a failure if some maximum number of iterations (e.g., 200 or 1000) occurs without successful decoding. When there is a failure, the partial decoding $\hat{\boldsymbol{x}}$ may serve as a useful starting point for another decoding algorithm [47].

We note in passing the difference between this decoding procedure and the widespread practice in the turbo code community, where the decoding algorithm is run for a fixed number of iterations (irrespective of whether the decoder finds a consistent state at some earlier time). This practice is wasteful of computer time. In our procedure, "undetected" errors would only occur if the decoder found an $\hat{\boldsymbol{x}}$ satisfying $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{z} \bmod 2$ which was not equal to the true $\boldsymbol{x}$. "Detected" errors occur if the algorithm runs for the maximum number of iterations without finding a valid decoding. The alternative practice mentioned above blurs this distinction between undetected and detected errors. Undetected errors are of scientific interest because they reveal distance properties of a code and in engineering practice, it would seem preferable for the detected errors to be labeled as erasures if practically possible.

B. Relationship to Other Algorithms

Meier and Staffelbach [47] implemented an algorithm similar to this sum-product decoder, also studied by Mihaljević and Golić [49], [50]. The main difference in their algorithm is that they did not distinguish between the probabilities q_{ml}^0 and q_{ml}^1 for different values of m; rather, they computed q_l^0 and q_l^1, as given above, and then proceeded with the horizontal step with all q_{ml}^0 set to q_l^0 and all q_{ml}^1 set to q_l^1.

Another related algorithm is the variational free-energy minimization decoder [37]. We describe the application of this decoder to Gallager and MN codes in Appendix C. Its performance is not as good as the sum-product decoder's.

C. Analysis of Decoding Algorithms

We analyze a simple decoding algorithm, following Gallager [27] and Meier and Staffelbach [47]. (The same algorithm has been used by Spielman [60].) We also study the sum-product decoder in the limit of large N using Monte Carlo methods. Although an explicit positive statement of the sum-product decoder's capabilities remains elusive, the important message of our analysis that follows is that the algorithms have "correction effects" which are independent of the blocklength N for large N. These results lead us to the following conjecture.

Conjecture 1: Given a binary-symmetric channel with noise density f, there exist practical decoders for Gallager and MN codes with rates R close to capacity, that can achieve negligible probability of error, for sufficiently large N.

1) Analysis of Correction Effect in a Simple Decoder: We introduce a simple decoder to iteratively solve for $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \bmod 2$.

```
Simple decoder:
Set $\hat{\boldsymbol{x}} = 0$.
[$\star$]: Evaluate $\hat{\boldsymbol{z}} = \boldsymbol{z} + \boldsymbol{A}\hat{\boldsymbol{x}} \bmod 2$. This is the check pattern
    that remains to be explained.
if $\hat{\boldsymbol{z}} = 0$, end.
Evaluate the "vote" vector $\boldsymbol{A}^{\mathrm{T}}\hat{\boldsymbol{z}}$ (not $\bmod 2$), which counts,
    for each bit $l$, the number of unsatisfied checks to which
    it belongs. The bits of $\boldsymbol{x}$ that get the most votes are viewed
    as the most likely candidates for being wrong. The biggest
    possible vote that a bit can receive is $t$, since each bit
    participates in $t$ checks.
Flip all bits of $\hat{\boldsymbol{x}}$ that have the largest vote.
go to [$\star$].
```

This decoding algorithm is not guaranteed to reach a stable state, but it is easy to analyze whether the first iteration of the decoder produces a "correction effect" or not. We say that there is a correction effect if the Hamming distance between $\hat{\boldsymbol{x}}$ and the true vector $\boldsymbol{x}$ decreases. We expect this decoding algorithm to have similar properties to those of the sum-product algorithm—at least for the first iteration—because, in the case of a binary-symmetric channel, the vote for a bit is directly related to its pseudoposterior probability.

Expected change in Hamming distance. We will assume that the weight of each row of $\boldsymbol{A}$ is an integer t_r, i.e., that t_r bits participate in each check. We consider the flipping of a single bit whose vote is t (the largest possible vote), and evaluate the probability that this reduces the Hamming distance. If the expected change in Hamming distance is negative, then we might hope that decoding algorithms of this type would work.

Consider a bit x_l that has value 0, and a check m in which it participates (i.e., an m such that $A_{ml} = 1$). The probability that $z_m = 1$, given that x_l is 0, is $p_z(t_r - 1)$, as defined in Section II-E2. So the probability that a 0 bit receives vote t is the binomial probability

$$a = P(\mathrm{vote}_l = t \,|\, x_l = 0) = [p_z(t_r - 1)]^t. \tag{54}$$

Similarly, the probability that $z_m = 1$, given that x_l is 1, is $1 - p_z(t_r - 1)$. So the probability that a 1 bit receives vote t is

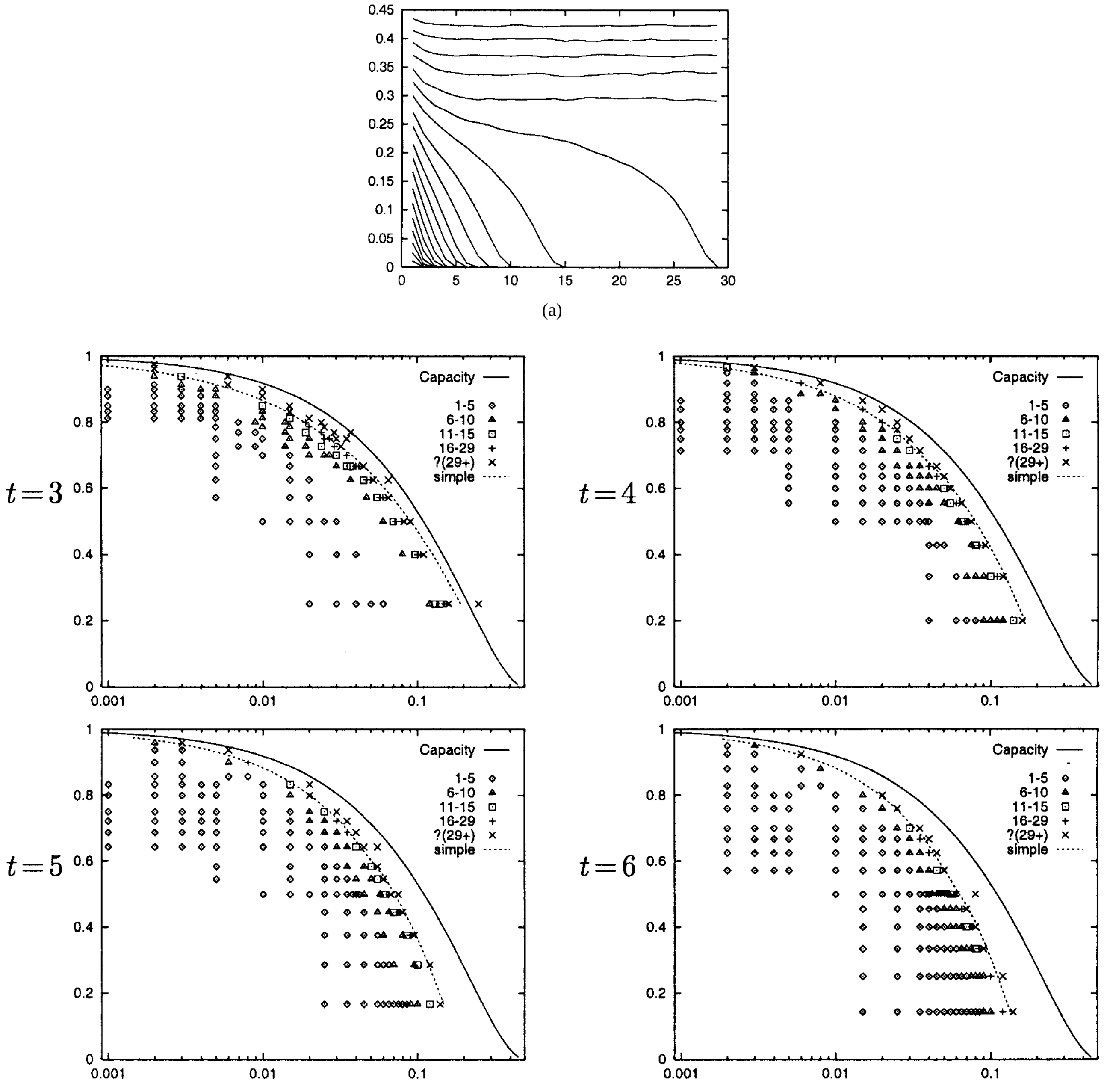
$$b = P(\mathrm{vote}_l = t \,|\, x_l = 1) = [1 - p_z(t_r - 1)]^t. \tag{55}$$

Now, given that a bit has vote t, the probabilities that it is a 0 or a 1 bit are, by Bayes' theorem

$$\begin{aligned} P(x_l = 0 \,|\, \mathrm{vote} = t) &\propto a(1-f) \\ P(x_l = 1 \,|\, \mathrm{vote} = t) &\propto bf. \end{aligned} \tag{56}$$

Thus the expected change in Hamming distance when a bit with vote t is flipped is

$$\frac{a(1-f) - bf}{a(1-f) + bf}. \tag{57}$$



(a)

(b)

Fig. 6. Analysis of practical decoding algorithms. (a) Time course of the decoding process in an infinite belief network with $t = 4$, $t_r = 8$. Graph shows average entropy of a bit as a function of number of iterations, as estimated by a Monte Carlo algorithm using 10 000 samples per iteration. Density f increases by steps of 0.005 from bottom graph $(f = 0.010)$ to top graph $(f = 0.100)$. There is evidently a transition atabout $f = 0.075$, above which the algorithm cannot determine $\boldsymbol{x}$. (b) Rates achievable by Gallager codes according to two analyses, for $t = 3, 4, 5, 6$ as a function of noise level. **Dotted lines** show estimates of decodability based on the first iteration of the simple decoder. Below the dotted line there is a correction effect; above the line, this decoder gives no correction effect. **Points** show theoretical success of decoding in infinite belief networks with various values of $t = 3, 4, 5, 6$, computed by Monte Carlo simulations of up to 29 iterations, as in part (a). Point styles diamond, square, triangle, and plus represent values of (f, R) at which complete decoding occurred after a number of iterations less than 29. Point style × denotes no decoding after 29 iterations.

If this is negative then there is a correction effect. We assume, rather simplisticly, that if there is no correction effect on the first iteration then reliable communication is not achievable using the code; that is, we assume that this analysis predicts a *bound* on the achievable communication rate.

For any given t and t_r (and hence λ), we can locate numerically the f above which there is no correction effect. The lower dotted curves in Fig. 6(b) show the corresponding information rate $(1 - 1/\lambda)$ of a Gallager code as a function of f for a variety of values of t and t_r, compared with the capacity. It is interesting that (at least for $f > 0.02$) as t increases, the bound on the achievable rate given by this simple analysis decreases. Thus in contrast to the results for the optimal decoder, where large t is best, we find that codes with small t $(t = 3)$ are best for practical decoding.

2) Analysis of Decoding of Infinite Networks by Monte Carlo Methods: The sum-product decoder's performance can be analyzed in terms of the decoding properties of an infinite

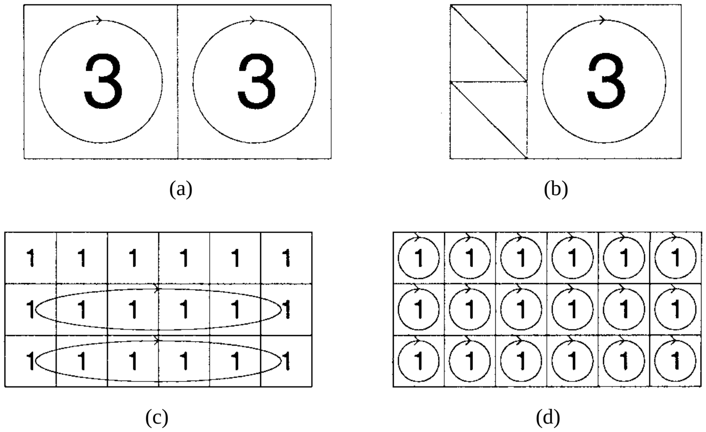


Fig. 7. Schematic illustration of (a) Construction 1A for a Gallager code with $t = 3$, $t_r = 6$, and $R = 1/2$, (b) Construction 2A for a Gallager code with rate $1/3$, (c), (d) two constructions similar to Construction 1A. Notation: an integer represents a number of permutation matrices superposed on the surrounding square. A diagonal line represents an identity matrix. A rotating ellipse imposed on a block within the matrix represents random permutation of all the columns in that block.

network without cycles (of which Fig. 5(g) shows a fragment). The larger the matrix $\boldsymbol{A}$, the closer its decoding properties should approach those that we now derive.

We consider an infinite belief network with no loops, in which every bit x_l connects to t checks and every check z_m connects to t_r bits. We consider the iterative flow of information in this network, and examine the average entropy of one bit as a function of number of iterations. At each iteration, a bit has accumulated information from its local network out to a radius equal to the number of iterations. Successful decoding will only occur if the average entropy of a bit decreases to zero as the number of iterations increases.

We have simulated the iteration of an infinite belief network by Monte Carlo methods—a technique first used by Gallager [27]. Imagine a network of radius I (the total number of iterations) centred on one bit. Our aim is to compute the conditional entropy of the central bit x given the state $\boldsymbol{z}$ of all checks out to radius I. To evaluate the probability that the central bit is 1 given a *particular* observation $\boldsymbol{z}$ involves an I-step propagation from the outside of the network into the center. At the ith iteration probabilities r at radius $I-i+1$ are transformed into q's and then into r's at radius $I-i$ in a way that depends on the states x of the unknown bits at radius $I-i$. In the Monte Carlo method, rather than simulating this network exactly, which would take a time that grows exponentially with I, we create for each iteration a representative sample (of size 100, say) of the values of $\{r, x\}$. In the case of a regular network with parameters t, t_r, each new pair $\{r, x\}$ in the list at the ith iteration is created by drawing the new x from its distribution and drawing at random with replacement $(t-1)(t_r-1)$ pairs $\{r, x\}$ from the list at the $(i-1)$th iteration; these are assembled into a tree fragment and the sum-product algorithm is run to find the new r value associated with the new node.

As an example, the results of runs with $t = 4$, $t_r = 8$, and noise densities f between 0.01 and 0.10, using 10 000 samples at each iteration, are shown in Fig. 6(a). It can be seen that runs with low enough noise level collapse to zero entropy after a small number of iterations, and those with high noise level decrease to a stable entropy level corresponding to a failure to decode.

The results of many such runs with various values of t and t_r are summarised by the points in Fig. 6(b), where the time to reach virtually zero entropy is indicated by the point style on a graph of rate versus noise level of the corresponding Gallager code, with × denoting "no decoding after 29 iterations." Regions in the graph where there are points corresponding to 1–29 iterations define values of the crossover probability f and rate R such that successful communication is possible with Gallager codes, according to the Monte Carlo analysis.

The two analyses, using the simple decoder and using Monte Carlo methods, appear to give similar predictions as to the maximum achievable rate R as a function of f.

We note that when the entropy drops to zero the decoding error probability falls to zero with a terminal convergence that is faster than exponential. The vertical step involves the multiplication of $t-1$ probabilities, so we believe the probability of decoding error falls asymptotically as $\exp(-a(t-1)^T)$ where T is the number of decoding iterations and a is a constant.

IV. Experimental Results Using the Sum-Product Decoder

Choice of ensemble. In early experiments using matrices from ensemble 4) of Section II-A, we examined some of the decoding errors made by the free-energy minimization decoder to be described in Appendix C and found that they tended to occur when the vector $\boldsymbol{x}$ was such that another slightly different typical vector $\boldsymbol{x}'$ had a similar (but not identical) encoding $\boldsymbol{z}'$. In terms of the random walk on the hypercube (Section II-C), these errors correspond to walks that after a small number of steps return close to the starting corner. They were possible because of rare topologies in the bipartite graph corresponding to the $\boldsymbol{A}$ matrix such as the topology illustrated in Fig. 5(c). We can eliminate the possibility of these errors by modifying the ensemble of matrices $\boldsymbol{A}$ so that the corresponding graph does not have short cycles in it. We made new $\boldsymbol{A}$ matrices by taking matrices from Ensemble 4 and deleting columns until there were no short loops of the type shown in Fig. 5(e). These matrices, having fewer columns,

TABLE I
RESULTS OF SUM-PRODUCT DECODING EXPERIMENTS FOR TWO GALLAGER CODES ON BINARY-SYMMETRIC CHANNELS. "trials" = NUMBER OF BLOCKS DECODED; "ers" = NUMBER OF BLOCK ERRORS. $p_{\rm ML}$ = MAXIMUM-LIKELIHOOD ESTIMATE OF *BLOCK* ERROR PROBABILITY. p_+ IS UPPER ERROR BAR FOR BLOCK ERROR PROBABILITY (APPENDIX B)

N	K	f	R	C	ers/trials	$\overline{\text{itns.}}$	$p_{\rm ML}$	p_+
19839	9839	.077	.496	.609	6/20603	19.5	.000291	.000659
		.076	.496	.612	3/114711	17.6	2.62e-05	8.3e-05
13298	3296	.155	.248	.378	3/2685	21.8	.00112	.00354
		.154	.248	.380	6/64543	20.7	9.3e-05	.00021
		.153	.248	.383	7/107635	19.3	6.5e-05	.000138
		.152	.248	.385	0/100809	18.1	0	1.98e-05
		.150	.248	.390	0/97058	16.4	0	2.06e-05

correspond to codes with slightly lower rates. They also have nonuniform weight per row.

We report results for codes with a variety of rates whose low density parity-check matrices are created in the following ways.

Construction 1A. An M by N matrix (M rows, N columns) is created at random with weight per column t (e.g., $t = 3$), and weight per row as uniform as possible, and overlap between any two columns no greater than 1. Fig. 7(a) shows the construction schematically for a rate $1/2$ code with $t = 3$.

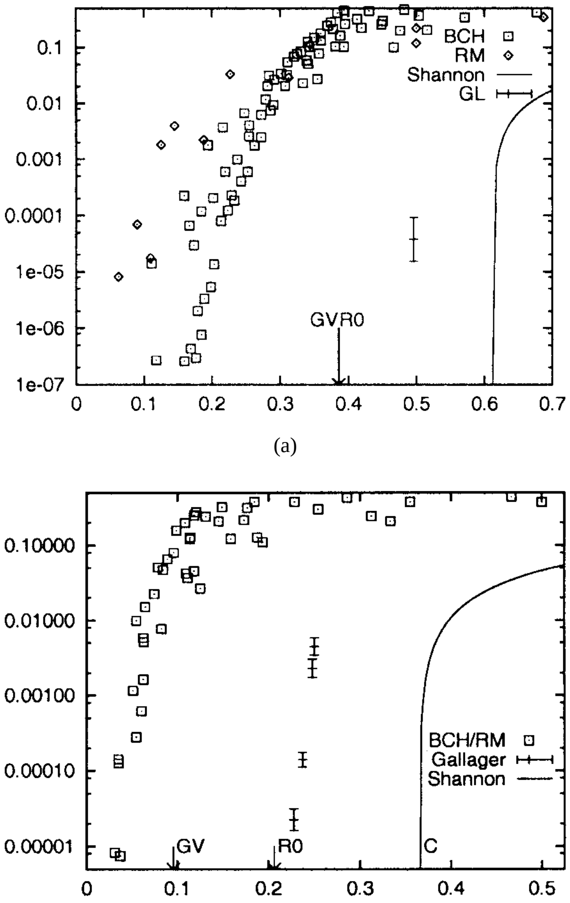
Construction 2A. Up to $M/2$ of the columns are designated weight 2 columns, and these are constructed such that there is zero overlap between any pair of columns. The remaining columns are made at random with weight 3, with the weight per row as uniform as possible, and overlap between any two columns of the entire matrix no greater than 1. Fig. 7(b) shows the construction schematically for a rate $1/3$ code. (This irregular construction using weight 2 columns was introduced because we guessed that it might give better practical performance; we used $M/2$ such columns because this was the maximum number of weight 2 columns for which it was easy to make "good" matrices; if more than $M/2$ columns of weight 2 are introduced at random then there is a risk that the corresponding code will have low weight codewords.)

Construction 1B and 2B. A small number of columns are deleted from a matrix produced by Constructions 1A and 2A, respectively, so that the bipartite graph corresponding to the matrix has no short cycles of length less than some length l, here $l = 6$.

Another way of constructing regular Gallager codes is to build the matrix $\boldsymbol{A}$ from nonoverlapping random permutation matrices as shown in Fig. 7(c), (d). Fig. 7(c) shows the construction used by Gallager [26]. For practical purposes, codes constructed in these ways appear to have very similar properties to codes made with Construction 1A, as long as cycles of length 4 are forbidden.

A. Rate of Codes Defined by These Matrices

The above constructions do not ensure that all the rows of the matrix are linearly independent, so the $M \times N$ matrix created is the parity-check matrix of a linear code with rate *at least* $R \equiv K/N$, where $K = N - M$. We report results on the assumption that the rate is R. The generator matrix of the code can be created by Gaussian elimination.



(a)

(b)

Fig. 8. Performance of Gallager codes applied to binary-symmetric channel and decoded by sum-product decoder. Comparison of empirical decoding results with calculated performance of Reed–Muller codes (diamonds) and BCH codes (squares), and the Shannon limit. BSC's with (a) $f_{\rm n} = 0.076$ (b) $f_{\rm n} = 0.16$ are assumed. Arrows show the values of $R_0(f_{\rm n})$ and $GV(f_{\rm n})$ for the channels. (a) Horizontal axis: information rate R. Vertical axis: **block** error probability. Curve: Shannon limit on achievable (rate, bit error probability) values. Results shown are for a code of Construction 1B whose parity matrix has $M = 10\,000$ rows and $N = 19\,839$. The weight per column is $t = 3$. (b) A BSC with $f_{\rm n} = 0.160$ is assumed. Horizontal axis: information rate R. Vertical axis: **bit**-error probability. Results shown are for a code of Construction 1A whose parity matrix has $M = 10\,000$ rows and $13\,336$ columns, and for three codes derived from this code by shortening, i.e., deleting columns from the parity matrix. These codes have $N = 13\,298, 13\,119, 12\,955$. The weight per column is $t = 3$.

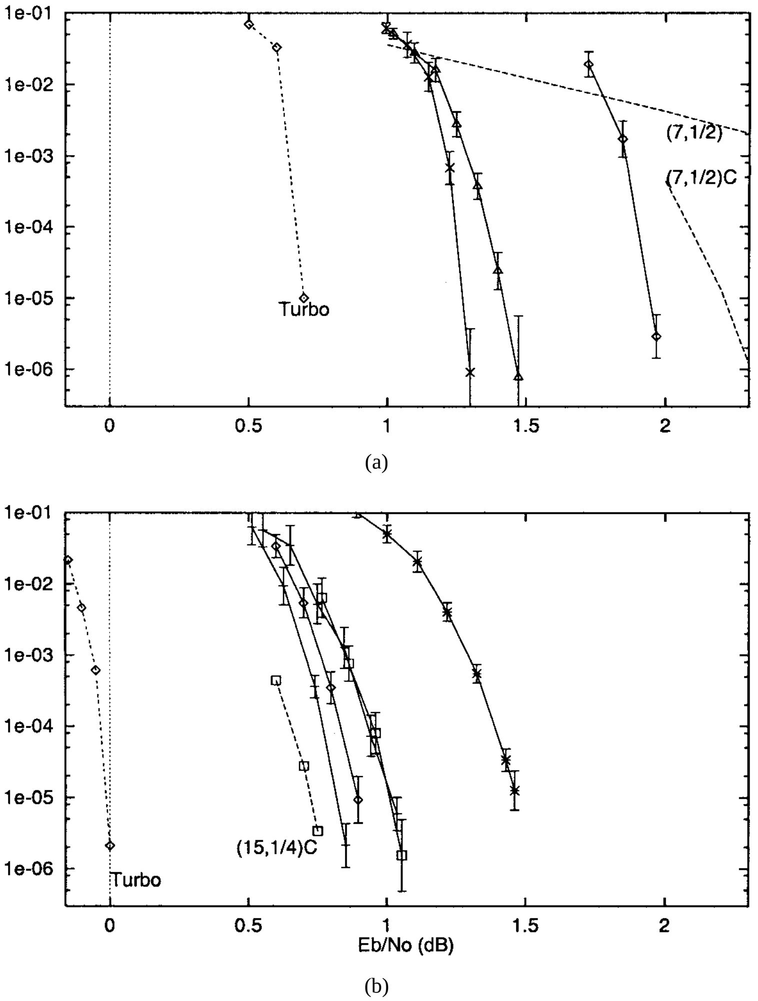


Fig. 9. Gallager codes' performance over Gaussian channel (solid curves) compared with that of standard textbook codes and state-of-the-art codes (dotted curves). Vertical axis shows empirical bit error probability. It should be emphasised that *all* the block errors in the experiments with Gallager codes were *detected* errors: the decoding algorithm reported the fact that it had failed. Panel (a) shows codes with rates between about $1/2$ and $2/3$; panel (b) shows codes with rates between $1/4$ and $1/3$. **Textbook codes:** The curve labeled $(7, 1/2)$ shows the performance of a rate $1/2$ convolutional code with constraint length 7, known as the *de facto* standard for satellite communications [29]. The curve (7,1/2)C shows the performance of the concatenated code composed of the same convolutional code and a Reed–Solomon code. **State of the art:** The curve (15,1/4)C shows the performance of an extremely expensive and computer intensive concatenated code developed at JPL based on a constraint length 15, rate $1/4$ convolutional code (data courtesy of R. J. McEliece.) The curves labeled Turbo show the performance of the rate $1/2$ Turbo code described in [12], [11] and the rate $1/4$ code reported in [21]. **Gallager codes:** From left to right the codes had the following parameters (N, K, R). Panel (a): $(65389, 32621, 0.499)$ (1B); $(19839, 9839, 0.496)$ (1B); $(29331, 19331, 0.659)$ (1B). Panel (b): $(40000, 10000, 0.25)$ (Construction 2A); $(29507, 9507, 0.322)$ (2B); $(14971, 4971, 0.332)$ (2B); $(15000, 5000, 0.333)$ (2A); $(13298, 3296, 0.248)$ (1B).

B. Empirical Results for Gallager Codes: Binary-Symmetric Channel

In the following experiments we performed up to 1000 iterations of the algorithm when decoding each $\boldsymbol{z}$, halting earlier if a consistent decoding was found. Most of the successful decodings took 20 iterations or fewer to be completed, which, for a code with blocklength $10\,000$, corresponds to a few seconds on a Sparc I workstation. We found that the results were best for $t = 3$ and became steadily worse as t increased.

We compare Gallager codes with $t = 3$ with Bose–Chaudhuri–Hocquenghem (BCH) codes, which are described in [52] as "the best known constructive codes" for memoryless noisy channels, and with Reed–Muller (RM) codes. These are multiple random error-correcting codes that can be characterized by three parameters (n, k, t). The blocklength is n, of which k bits are data bits and the remainder are parity bits. Up to t errors can be corrected in one block.

Fig. 8 compares two Gallager codes with BCH and RM codes on two binary-symmetric channels. To compute the probability of error for BCH codes we evaluated the probability of more than t errors in n bits. Similarly, for RM codes of minimum distance d, performance was computed assuming that more than $\lfloor d/2 \rfloor$ errors cannot be corrected. (See Appendix B for discussion of how the presented results were computed.)

The mean number of iterations of the algorithm to obtain a successful decoding is displayed for a selection of codes and different densities f in Table I. In rare cases as many as 800 iterations took place before a successful decoding emerged.

C. Gallager Codes for the Gaussian Channel

We originally conceived MN codes as codes for the memoryless binary-symmetric channel. It turns out, however, that

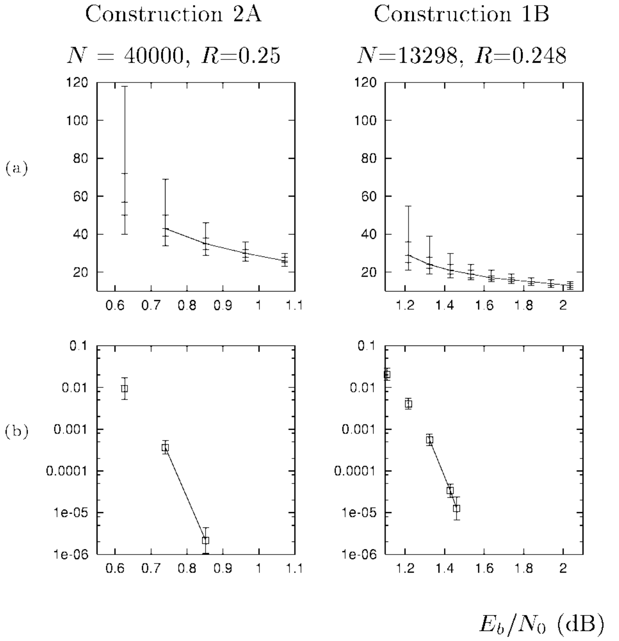


Fig. 10. (a) Median number of iterations of sum-product algorithm taken to obtain a successful decoding. Bars show 5th, 25th, 75th, and 95th percentiles. (b) Corresponding bit error probability.

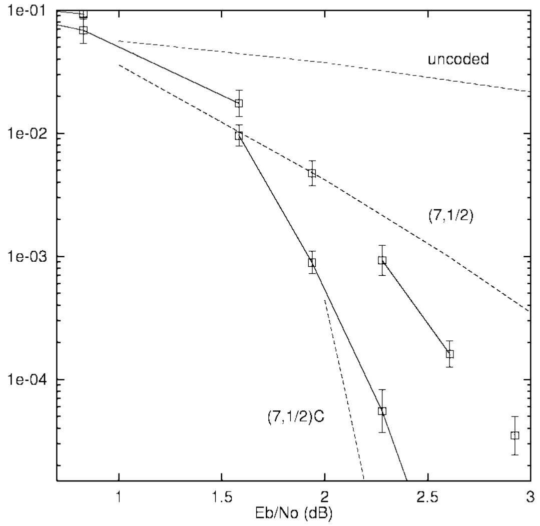


Fig. 11. Short-blocklength Gallager codes' performance over Gaussian channel (solid curves) compared with that of standard textbook codes (dotted curves). Vertical axis shows empirical bit error probability. It should be emphasised that *all* the block errors in the experiments with Gallager codes were *detected* errors: the decoding algorithm reported the fact that it had failed. **Textbook codes:** as in Fig. 9. **Gallager codes:** From left to right the codes had the following parameters (N, K, R): $(1008, 504, 0.5)$ (Construction 1A); $(504, 252, 0.5)$ (1A).

Gallager and MN codes have a much broader applicability. As we proved in Section II, Gallager codes are very good for any symmetric stationary ergodic noise model (including arbitrary correlations and memory—Definition 6) as long as a mean entropy can be defined for it. Here we report investigations of nonuniform noise corresponding to communication over a Gaussian channel with binary inputs.

Fig. 9 compares the performance of Gallager codes with rates between $1/4$ and $2/3$ with textbook codes and with state-of-the-art codes. As before, the best results are obtained by making the weight per column t as small as possible (Constructions 2A and 2B). Unsurprisingly, codes with large blocklength are better.

In some cases we modified the matrices so as to eliminate short cycles. The original matrices, by construction, had no cycles of length 4, a constraint which was found to be beneficial. We deleted columns so as to remove cycles of length $6, 8, \cdots$, expecting that this would further improve performance. However, we found that these modifications made little difference.

For the codes with blocklengths and $40\,000$ and $13\,298$ in Fig. 9 the median number of iterations taken to complete a successful decoding is shown in Fig. 10(a) as a function of E_b/N_0. The line shows the median number of iterations and the bars show the 5th, 25th, 75th, and 95th percentiles. Fig. 10(b) shows the corresponding bit error probabilities, reproduced from Fig. 9.

D. Results for Small Blocklengths

To double-check our results against Gallager's we replicated experiments with codes of blocklength 504 and 1008. Our random code constructions are not identical to Gallager's, and we ran the decoder for more iterations (up to 500), but the results for Construction 1A appear much the same as those in his book [27, fig. 6.7].

V. Pictorial Demonstration of Gallager Codes

Figs. 12–15 illustrate visually the conditions under which Gallager's low-density parity-check codes can give reliable communication over binary-symmetric channels and Gaussian channels. These demonstrations may be viewed as animations on the world wide web [39].

A. Encoding

Fig. 12 illustrates the encoding operation for the case of a Gallager code whose parity-check matrix is a $10\,000 \times 20\,000$ matrix with three 1's per column. The high density of the generator matrix is illustrated in (b) and (c) by showing the change in the transmitted vector when one of the $10\,000$ source bits is altered. Of course, the source images shown here are highly redundant, and such images should really be compressed before encoding. Redundant images are chosen in these demonstrations to make it easier to see the correction process during the iterative decoding. The decoding algorithm does *not* take advantage of the redundancy of the source vector, and it would work in exactly the same way irrespective of the choice of source vector.

B. Iterative Decoding

After the transmission is sent over a channel with noise level $f = 7.5\%$, the received vector is as shown in the upper left of Fig. 13. The subsequent pictures in Fig. 13 show the iterative probabilistic decoding process. The sequence of figures shows the best guess, bit by bit, given by the iterative decoder, after 0, 1, 2, 3, 10, 11, 12, and 13 iterations. The decoder halts

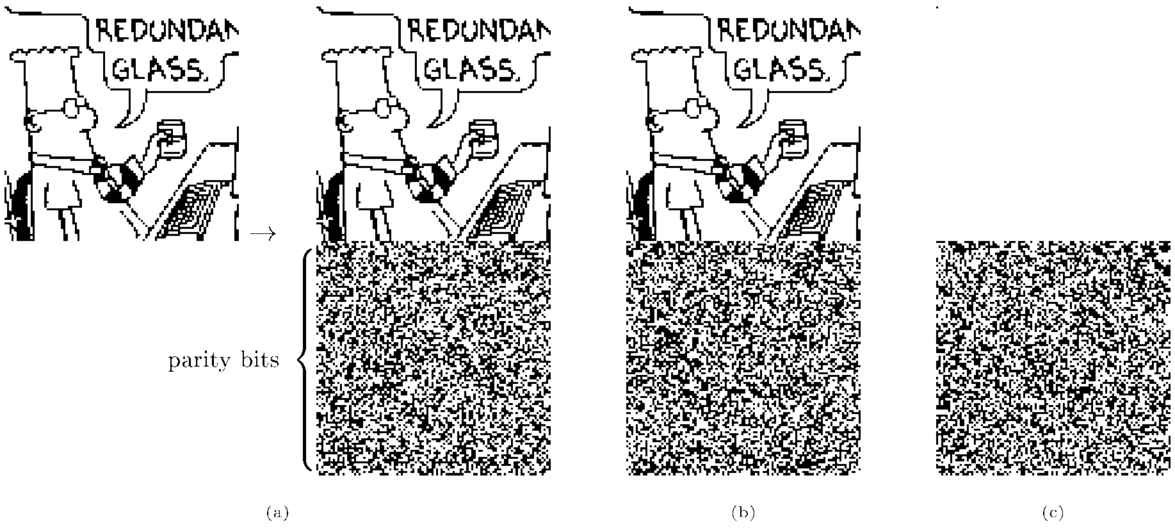


Fig. 12. Demonstration of encoding with a rate $1/2$ Gallager code. The encoder is derived from a very sparse 10000×20000 parity-check matrix with three 1's per column. (a) The code creates transmitted vectors consisting of 10 000 source bits and 10 000 parity-check bits. (b) Here, the source sequence has been altered by changing the first bit. Notice that many of the parity-check bits are changed. Each parity bit depends on about half of the source bits. (c) The transmission for the case $\boldsymbol{s} = (1, 0, 0, \cdots, 0)$. This vector is the difference (modulo 2) between transmissions (a) and (b). (Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.)

after the 13th iteration when the best guess violates no parity checks. This final decoding is error free.

C. Gaussian Channel

In Fig. 14 the first picture shows the transmitted vector and the second shows the received vector after transmission over a Gaussian channel with $x/\sigma = 1.185$. The greyscale represents the value of the normalized likelihood, $\frac{P(y|t=1)}{P(y|t=1)+P(y|t=0)}$. This signal-to-noise ratio $x/\sigma = 1.185$ is a noise level at which this rate $1/2$ Gallager code communicates reliably (the probability of error is $\simeq 10^{-5}$). To show how close we are to the Shannon limit, the third panel shows the received vector when the signal-to-noise ratio is reduced to $x/\sigma = 1.0$, which corresponds to the Shannon limit for codes of rate $1/2$.

Fig. 15 shows the analogous vectors in the case of a code with rate about $1/4$.

VI. MN Codes

A. The Ideas Behind MN Codes

It is conventional to define a linear error-correcting code to have transmitted blocklength $N > K$, and to use signals $\boldsymbol{s}$ of density $f_{\rm s} = 0.5$. Conventionally, the code is systematic, so the first K transmitted bits are the K source bits. The $(N-K)$ extra bits are parity-check bits, which produce redundancy in the transmitted vector $\boldsymbol{t}$. This redundancy is exploited by the decoding algorithm to infer the noise vector $\boldsymbol{n}$.

MN codes [40] are based on a different approach. We first assume that the source may itself be redundant, having $f_{\rm s}$, the expected density of $\boldsymbol{s}$, 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 [70], [57]; one simply reverses the role of encoder and decoder in a standard arithmetic coder based on a model corresponding to the sparse messages (see Appendix G). Now 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, $\rho \equiv K/N$, and the "information rate" of the code, $R \equiv H_2(f_{\rm s})K/N$. Error-free communication may be possible if the information rate is less than the capacity of the channel. For example, consider a binary-symmetric channel having $f_{\rm n} = 0.1$, and assume that we have a source with density $f_{\rm s} = 0.1$. Then we might construct a code with $N = K$, i.e., a square linear code with symbol rate 1 [Fig. 16(b)]. 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 key idea behind MN codes is that we construct the generator matrix in terms of an *invertible* matrix, in such a way that the sparse source and the sparse noise can be treated symmetrically in the decoding problem—in contrast to conventional syndrome decoding where only the noise vector appears in the problem.

1) Code Construction: MN codes make use of the same matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ that were constructed in Section I-C1. These matrices will now be denoted by $\boldsymbol{C}_{\rm n} \equiv \boldsymbol{C}_2$ (the square invertible matrix) and $\boldsymbol{C}_{\rm s} \equiv \boldsymbol{C}_1$. We redefine N and $K = \rho N$ such that $\boldsymbol{C}_{\rm s}$ is an $N \times K$ matrix and $\boldsymbol{C}_{\rm n}$ is an $N \times N$ matrix.

2) Encoding: A source vector $\boldsymbol{s}$ of length ρN is encoded into a transmitted vector $\boldsymbol{t}$ defined by (Fig. 17(a))

$$\boldsymbol{t} = \boldsymbol{C}_{\rm n}^{-1}\boldsymbol{C}_{\rm s}\boldsymbol{s} \bmod 2. \tag{58}$$

This encoding operation takes time of order $\min[\rho N t + N^2, \rho N^2]$. The mapping from source bits to transmitted bits is a linear mapping, however, MN codes are *nonlinear* codes in the sense that the codewords that have high probability do not form a complete linear subspace of $\{0,1\}^N$.

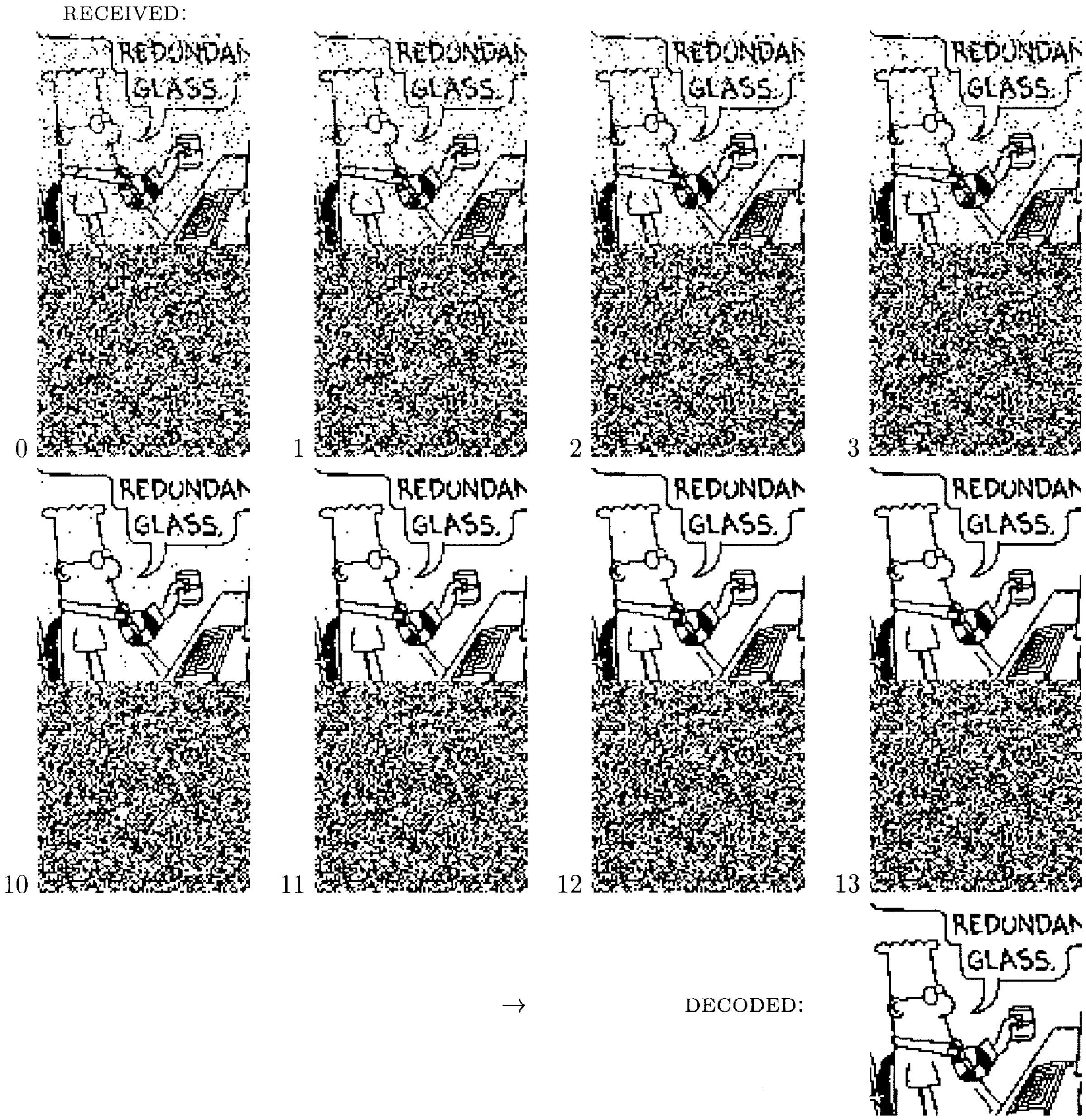


Fig. 13. Iterative probabilistic decoding of a Gallager code. The sequence of figures shows the best guess, bit by bit, given by the iterative decoder, after 0, 1, 2, 3, 10, 11, 12, and 13 iterations. The decoder halts after the 13th iteration when the best guess violates no parity checks. This final decoding is error free. (Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.)

3) The Decoding Problem: The received vector is

$$\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{n} \bmod 2 \tag{59}$$

where the noise $\boldsymbol{n}$ is assumed to be a sparse random vector with independent and identically distributed bits, density $f_{\rm n}$. (See Appendix A for discussion of other channels.)

The first step of the decoding is to compute

$$\boldsymbol{z} = \boldsymbol{C}_{\rm n}\boldsymbol{r} \bmod 2 \tag{60}$$

which takes time of order Nt.

Because

$$\boldsymbol{z} = \boldsymbol{C}_{\rm n}(\boldsymbol{t} + \boldsymbol{n}) \bmod 2 = \boldsymbol{C}_{\rm s}\boldsymbol{s} + \boldsymbol{C}_{\rm n}\boldsymbol{n} \bmod 2$$

the decoding task is then to solve for $\boldsymbol{x} = [\begin{smallmatrix}\boldsymbol{s}\\ \boldsymbol{n}\end{smallmatrix}]$ the equation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \bmod 2 \tag{61}$$

where $\boldsymbol{A}$ is the N by $K + N$ matrix $[\boldsymbol{C}_{\rm s}|\boldsymbol{C}_{\rm n}]$. This decoding problem is shown schematically in Figs. 17(b) and 18 for MN codes with symbol rates $\rho = 1$ and $1/3$.

We emphasize two properties of (61).

1) There is a pleasing symmetry between the sparse source vector $\boldsymbol{s}$ and the sparse noise vector $\boldsymbol{n}$, especially if $f_{\rm s} = f_{\rm n}$.
2) Both the matrix $\boldsymbol{A}$ and the unknown vector $\boldsymbol{x}$ are sparse (the bits of $\boldsymbol{x}$ have density $f_{\rm s}$ or $f_{\rm n}$), so the decoding problem is identical to the syndrome decoding problem for Gallager codes.

B. Theoretical Results for MN Codes

The theoretical properties of optimal decoding derived in Section II imply that good MN codes exist. Fig. 19 shows

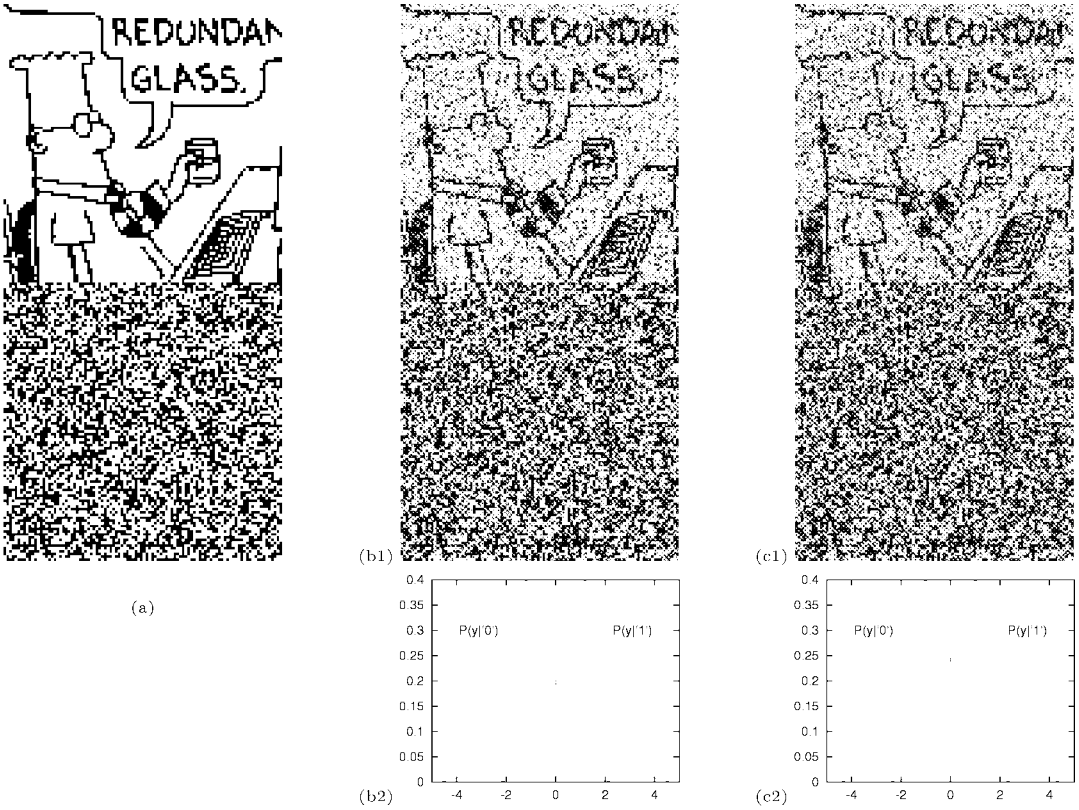


Fig. 14. Demonstration of a Gallager code for a Gaussian channel. (a) A data transmission consisting of 10 000 source bits and 10 000 parity-check bits. (b1) The received vector after transmission over a Gaussian channel with $x/\sigma = 1.185$. ($E_b/N_0 = 1.47$ dB.) The greyscale represents the value of the normalized likelihood. This transmission can be perfectly decoded by the sum-product decoder, that is, the decoder's output is identical to the original data shown in (a). The empirical probability of decoding failure is about 10^{-5}. (b2) The probability distribution of the output y of the channel with $x/\sigma = 1.185$ for each of the two possible inputs. (c1) The received transmission over a Gaussian channel with $x/\sigma = 1.0$, which corresponds to the Shannon limit. (c2) The probability distribution of the output y of the channel with $x/\sigma = 1.0$ for each of the two possible inputs. (Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.)

the communication rates proved achievable with MN codes communicating over a binary-symmetric channel with $f_{\mathrm{n}} = f_{\mathrm{s}}$. This figure was produced by the same method as Fig. 3.

C. Experimental Results: One MN Code Can be Used for Channels with a Range of Noise Levels

1) Binary-Symmetric Channel: We initially made experiments in which a sparse source communicated over a binary-symmetric channel, with $f_{\mathrm{s}} = f_{\mathrm{n}}$. Results for two codes with symbol rates about 1 and about $1/3$ are shown in Table II and Fig. 20(a).

We then explored cases with $f_{\mathrm{s}} > f_{\mathrm{n}}$, *using the same encoder* with $K = 3296$ and $N = 10002$. Fig. 20 shows that a single encoder can be used to transmit quite near to capacity over two channels with substantially different noise levels (15.3% and 11.4%), simply by changing the density of the source stream. Contrary to our expectations, the performance appeared to get better when the symmetry between the source and the noise in the decoding was broken; in the case with $f_{\mathrm{s}} = 0.5$ and $f_{\mathrm{n}} = 0.114$, the performance is beyond R_0.

2) MN Codes for the Gaussian Channel: We have simulated MN codes with dense sources and sparse sources. The rate 0.33 code with a dense source gave a probability of bit error less than 10^{-5} at $E_b/N_0 = 1.81$ dB. The same code, encoding a sparse source with $f_{\mathrm{s}} = 0.2$ $(R = 0.238)$ gave a similar error probability at $E_b/N_0 = 2.31$ dB. MN codes seem to be inferior to Gallager codes in terms of E_b/N_0, but it may be that their novel properties offer compensating benefits.

VII. Discussion

This paper has given a semiconstructive proof of the noisy channel coding theorem using low-density parity-check codes. Gallager and MN codes are good not only for the binary-symmetric channel but also for any channel models for which the optimizing input distribution is symmetrical and the law of large numbers holds. It is a surprise to us that a single code can be good for any channel. We had anticipated that to achieve very good performance on a new channel (such as a bursty noise channel), a new custom-designed code would be needed. This expectation is shared by Golomb, Peile, and Scholtz, who state that "the optimal code for a given set of channel conditions may not resemble the optimal code for another" [29, p. 369]. But theoretically, the same encoder family can be used for *any* channel—all that needs to be changed is the decoding algorithm.

The practical performance of Gallager's 1962 codes, using Gallager's 1962 decoding algorithm, would have broken practical coding records up until 1993. The decoder works

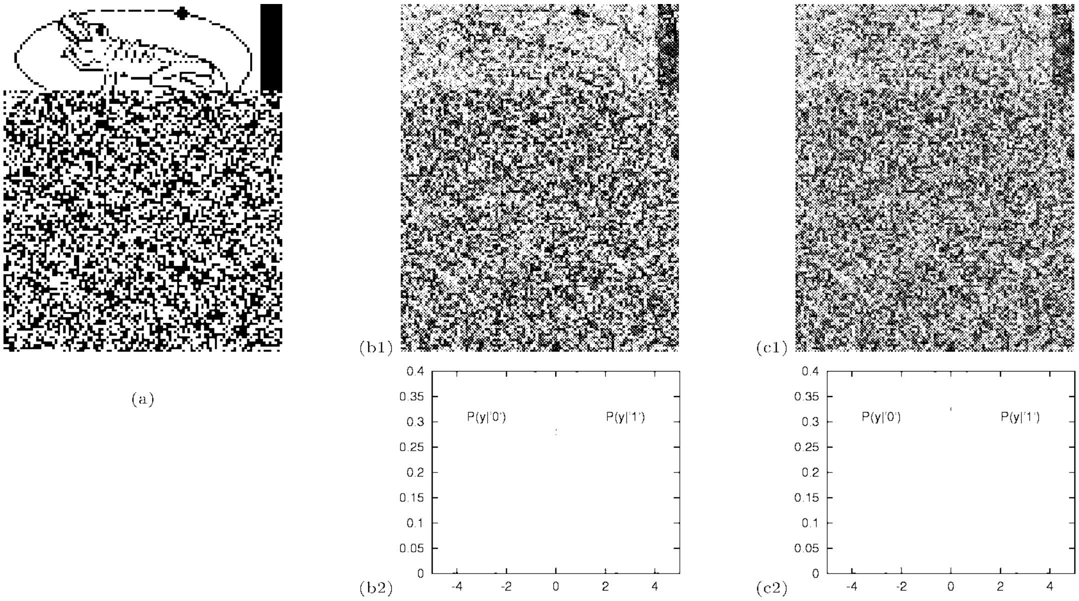


Fig. 15. Demonstration of a rate $1/4$ Gallager code for a Gaussian channel. (a) A data transmission consisting of 3296 source bits and 10 002 parity check bits. (b1) The received vector after transmission over a Gaussian channel with $x/\sigma = 0.84$. ($E_b/N_0 =$ 1.5 dB.) This transmission can be perfectly decoded by the sum-product decoder, that is, the decoder's output is identical to the original data shown in (a). The empirical probability of decoding failure is about 10^{-5}. (b2) The probability distribution of the output y of the channel with $x/\sigma = 0.84$ for each of the two possible inputs. (c1) The received transmission over a Gaussian channel with $x/\sigma = 0.64$, which corresponds to the Shannon limit. (b2) The probability distribution of the output y of the channel with $x/\sigma = 0.64$ for each of the two possible inputs. The crocodile image is the insignia of the Cavendish Laboratory.

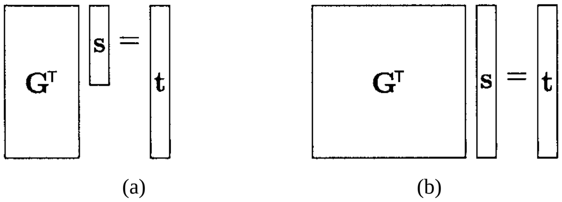


Fig. 16. (a) A conventional code. The source vector $\boldsymbol{s}$, of length K, is dense. The transmitted vector $\boldsymbol{t}$ is of length $N > K$. Here $N = 2K$, so the symbol rate and information rate are both $K/N = 0.5$ shannons. (b) Square code for a sparse source, having $N = K$. The symbol rate ρ is 1, but if the density of the source $f_{\rm s}$ is 0.1 then the information rate is $H_2(0.1) \simeq 0.5$ shannons, the same as that of the conventional code.

beyond the minimum distance of the code, beyond the Gilbert bound, and beyond the rate that was widely believed to be the "effective capacity" R_0.

As far as we know, the only traditional code that can match the performance of Gallager codes is the code for Galileo developed at JPL, which employs a rate $1/4$, constraint length 15 convolutional code surrounded by a Reed–Solomon code, giving an effective blocklength of about 8000 bits (R. J. McEliece, personal communication). This system can only be decoded using expensive special-purpose hardware, and the details of the code are unpublished outside JPL [61].

A. Comparison with Turbo Codes

We heard about turbo codes [12], [11], which outperform Gallager's codes in terms of E_b/N_0, towards the end of this work. There are some similarities between the codes. The turbo decoding algorithm may be viewed as a sum-product algorithm ([69], [68], [46]). Turbo codes also have a construction in terms of sparse random trellises. Indeed, as

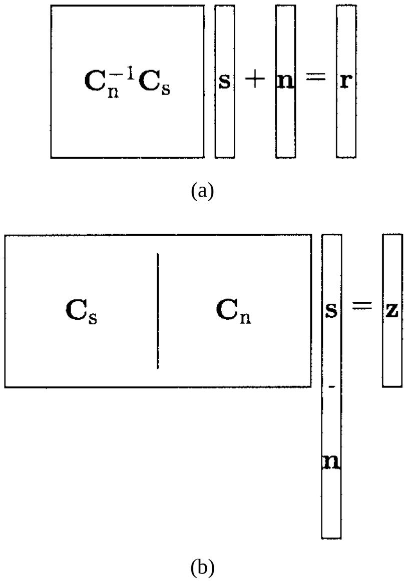


Fig. 17. Pictorial representation of MN Code with symbol rate $\rho = 1$. (a) Encoding, transmission and reception. The vectors $\boldsymbol{s}$ and $\boldsymbol{n}$ are sparse. The matrices $\boldsymbol{C}_{\rm s}$ and $\boldsymbol{C}_{\rm n}$ are very sparse. (b) Decoding. The vector $\boldsymbol{z}$ is given by $\boldsymbol{z} = \boldsymbol{C}_{\rm n}\boldsymbol{r} \bmod 2$. We attempt to solve for $\boldsymbol{s}$ and $\boldsymbol{n}$.

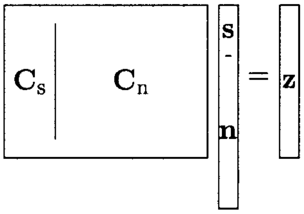


Fig. 18. The decoding situation for an MN code with symbol rate $\rho = 1/3$.

TABLE II

RESULTS OF SUM-PRODUCT DECODING EXPERIMENTS FOR TWO MN CODES ON BINARY-SYMMETRIC CHANNELS. "trials" = NUMBER OF BLOCKS DECODED; "ers" = NUMBER OF BLOCK ERRORS. p_{ML} = MAXIMUM-LIKELIHOOD ESTIMATE OF BLOCK ERROR PROBABILITY. p_+ IS UPPER ERROR BAR FOR BLOCK ERROR PROBABILITY (APPENDIX B). HERE, $f_{\mathrm{s}} = f_{\mathrm{n}} = f$. SEE ALSO FIG. 20(a)

N	K	t	f	R	C	ers/trials	itns.	p_{ML}	p_+
10000	9839	3	.077	.385	.609	6/20603	19.5	.000291	.000659
			.076	.382	.612	3/114711	17.6	2.62e-05	8.3e-05
10002	3296	3	.155	.205	.378	3/2685	21.8	.00112	.00354
			.154	.204	.380	6/64543	20.7	9.3e-05	.00021
			.153	.203	.383	7/107635	19.3	6.5e-05	.000138
			.152	.203	.385	0/100809	18.1	0	1.98e-05
			.150	.201	.390	0/97058	16.4	0	2.06e-05

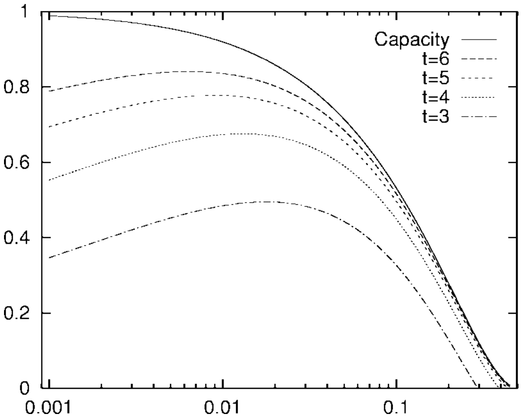


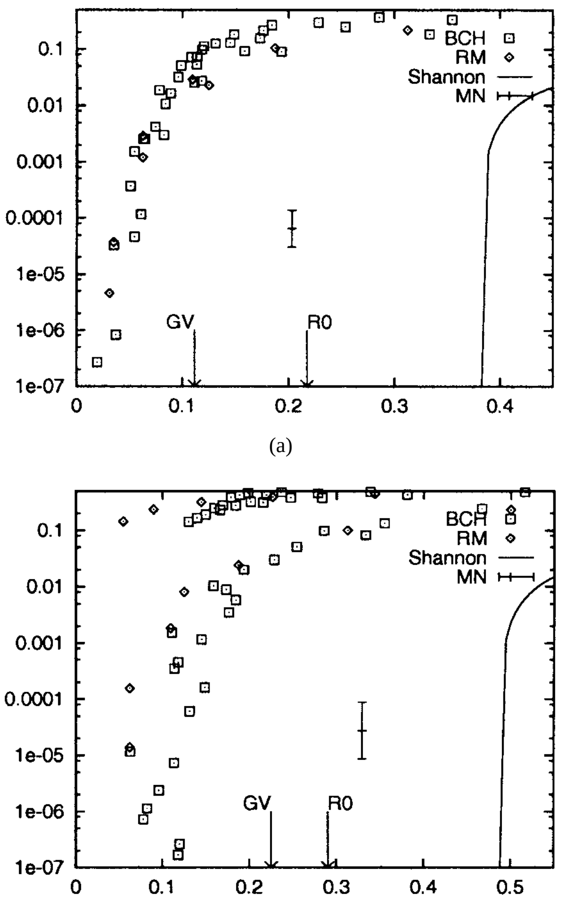
Fig. 19. Lower bounds on achievable information rate in shannons versus noise level f for MN codes with t from 3 to 6. The solid line shows the channel capacity. The lines are lower bounds on rates achievable by MN codes. This achievable region was obtained using the first ensemble of matrices $\boldsymbol{A}$. As the weight per column t increases the achievable region rises towards the fundamental limit, the capacity.

shown schematically in Fig. 21, turbo codes *are* low-density parity-check codes. However, few "goodness" properties have been proved for turbo codes.

At error probabilities of about 10^{-5}, Gallager and MN codes are not able to get quite so close to the Shannon limit as turbo codes (Fig. 9). However, turbo codes as originally presented are known to have an error "floor" at about 10^{-6} due to low-weight codewords (B. J. Frey, personal communication); the error probability of these turbo codes no longer decreases rapidly with increasing E_b/N_0 below this floor. We have seen no evidence of such a floor in Gallager codes; and theoretically we do not expect Gallager codes to have the low-weight codewords that could give rise to this behavior. So it is possible that at very low-bit error probabilities, Gallager codes outperform turbo codes. It should also be emphasised that all the errors made by Gallager codes that we have observed are *detected* errors, whereas the turbo codes's errors that are caused by low-weight codewords are *undetected* errors. Gallager codes may also have an advantage over turbo codes in terms of their decoding complexity.

B. Computational Complexity

In a brute-force approach, the time to create a Gallager code scales as N^3, where N is the block size. The encoding time scales as N^2, but encoding involves only binary arithmetic, so for the blocklengths studied here it takes considerably less time than the simulation of the Gaussian channel. We are currently investigating the performance of low-density parity-check codes which can be encoded in linear time [42].

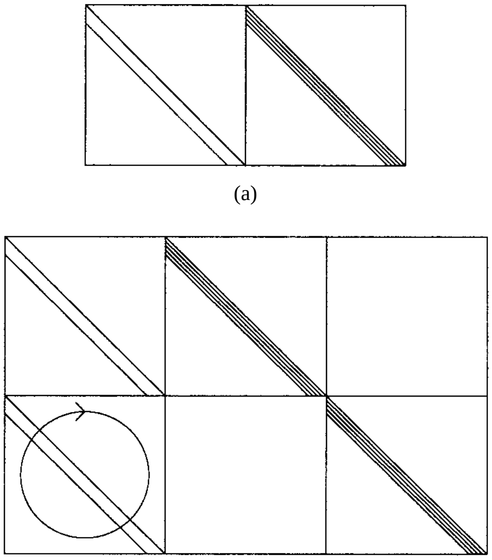


(a)

(b)

Fig. 20. *One* MN code with fixed generator matrix can communicate at good rates over *two* binary-symmetric channels with substantially different noise levels by changing the source density. In (a) $f_{\mathrm{n}} = 0.153$ and the source is sparse with density $f_{\mathrm{s}} = 0.153$; in (b) $f_{\mathrm{n}} = 0.114$, and a dense source is used ($f_{\mathrm{s}} = 0.5$). The empirical decoding results are compared with calculated performance of Reed–Muller codes (diamonds) and BCH codes (squares), and the Shannon limit (solid curve). Horizontal axis: information rate R. Vertical axis: **block** error probability ϵ. Arrows show the values of R_0 and $GV(f_{\mathrm{n}})$. The MN code has $N = 10000$, $K = 3296$, and $t = 3$ (Construction 1B).

Decoding involves approximately $6Nt$ floating-point multiplies per iteration (assuming a model of computation where



(a)

(b)

Fig. 21. Convolutional codes and turbo codes *are* low-density parity-check codes. Schematic pictures of the parity-check matrices of (a) a systematic recursive convolutional code with (binary) generator polynomials $(10001/11111)$ and (b) a rate $1/3$ turbo code formed by combining two such convolutional codes. Notation: A band of diagonal lines represent a band of diagonal 1's. Horizontal and vertical lines indicate the boundaries of the blocks within the matrix. In (a) the left-hand band is 10001, corresponding to the numerator polynomial, and the right-hand band, 11111, corresponds to the denominator. In (b) the first K bits are the systematic bits, the next K are the parity bits from the first convolutional code, and the last K are the parity bits from the second convolutional code, which receives the systematic bits in a permuted order. The weight per row of the turbo code's parity-check matrix is 7 for almost all rows, and the weight per column is 4 or 5.

the cost of elementary operations does not grow with N), so the total number of operations per decoded bit (assuming 20 iterations) is about $120t/R$, independent of blocklength. For the codes presented here, this is about 800 operations. This is not at all excessive when compared with textbook codes—the constraint length 7 convolutional code used by Voyager requires 256 operations per decoded bit. The turbo codes of [12] require about 3800 operations per decoded bit (B. J. Frey, personal communication).

Strictly, a constant number of iterations (taken above to be 20) is not sufficient to achieve *negligible* probability of error for any blocklength [26]. Assuming the truth of the conjecture of Section III-C2 that the bit error probability decreases as $\exp(-a(t-1)^T)$ where T is the number of decoding iterations and a is a constant, in order for this probability to decrease as $1/N$ with increasing N, we need the number of decoding iterations to grow as $T \sim \log\log N$.

The decoding algorithm involves no adjustable parameters, except those associated with the handling of overflows. After each vertical step we prevented all the probabilities from going greater than $1-10^{-10}$ or less than 10^{-10}. (One could view the "update schedule," i.e., the order in which the quantities q and r are updated, as an adjustable aspect of the algorithm [24]; we have not explored this option. We have briefly examined two modifications of the algorithm, making the prior probabilities more extreme if a decoding has not emerged, and making the propagated probabilities more (or less) extreme, but we have not found any useful improvement in performance. However, turbo code researchers have found similar tweaks to the sum-product algorithm are helpful [21].)

The encoding and decoding software and the parity check matrices used in this paper are available from http://wol.ra.phy.cam.ac.uk/mackay/codes/.

C. Descriptive Complexity

The descriptive complexity of these codes is much smaller than the descriptive complexity of arbitrary linear codes, which is $\sim NK$ bits. A Gallager (N, K) code has a descriptive complexity of about $tN\log(N-K)$ bits, since for every one of N columns we have to select t bits from $N-K$.

D. Distance Properties

We have proved minimum distance properties of Gallager codes in Section II-C2 (the Gilbert bound can be attained), but we do not view this as a primary result. We view distance properties as a secondary attribute compared with the block error probability. The minimum distance of a code may be viewed as a convenient measure of how "good" it is, but in fact it is not possible to distinguish between good and very good codes by their minimum distance, and bounded distance decoders are well known to be unable to achieve the Shannon limit [43]. We have proved that Gallager and MN codes can (when optimally decoded) achieve capacity. Moreover, we have demonstrated error correcting abilities at rates well above the Gilbert rate.

E. Discussion Specific to MN Codes

In a conventional linear (N, K) code, the codewords form a complete linear subspace of $\{0,1\}^N$, and it is conventional to assume that its generator matrix $\boldsymbol{G}$ might as well be put in systematic form. In designing MN codes we made the assumption instead that the source is *sparse*, so the codewords that have high probability are only a small subset of a complete linear subspace. In this sense, MN codes are nonlinear codes, even though the transmitted vector $\boldsymbol{t}$ is a linear function of a source vector $\boldsymbol{s}$. The generator matrix may *not* be put in systematic form. The systematic form for a code with symbol rate $\rho = 1$ would simply be an identity matrix, giving no error protection at all. We think the MN code's sparse source is an interesting idea which could have a variety of spinoffs. For example, MN codes offer the potentially useful property that the rate of the code can be changed without changing the generator matrix.

F. Application of MN Codes to Multiple-User Channels

Consider a multiple-user linear binary channel whose output, each cycle, is $r = t^{(1)} + t^{(2)} + \cdots + t^{(U)} + n \bmod 2$, where $t^{(u)}$ is the bit transmitted by user u, and n is noise. The information-theoretic bound on the total information that the U users can communicate is the capacity $C(f_{\rm n}) = 1 - H_2(f_{\rm n})$.

We can create multiple-user codes for this channel directly from MN codes that encode K sparse source bits into N transmitted bits. The K columns of the matrix $\boldsymbol{C}_{\rm s}$

are divided up between the users, with user u receiving K_u columns, forming a matrix $\boldsymbol{C}_{\rm s}^{(u)}$. All users know the sparse matrix $\boldsymbol{C}_{\rm n}$, and the decoder knows the entire matrix $\boldsymbol{A} = [\boldsymbol{C}_{\rm s}^{(1)}|\boldsymbol{C}_{\rm s}^{(2)}|\cdots|\boldsymbol{C}_{\rm s}^{(U)}|\boldsymbol{C}_{\rm n}]$. In each block cycle, user u encodes a vector $\boldsymbol{s}^{(u)}$ of K_u bits with density f_u into a vector of length N bits, $\boldsymbol{t}^{(u)} = \boldsymbol{C}_{\rm n}^{-1}\boldsymbol{C}_{\rm s}^{(u)}\boldsymbol{s}^{(u)}$, and transmits this vector.

The properties proved for MN codes immediately carry over to these multiple-user codes. In particular, the Shannon limit for the multiple-user linear channel can be achieved, given an optimal decoder. Such a system, if appropriately controlled, would allow the users dynamically to change their rate of communication by changing their densities f_u without changing their encoder.

G. Conundrum: Why Were Gallager Codes Forgotten?

Why was Gallager's work mostly forgotten by the information theory community?

There are very few citations of Gallager's [26], [27] work on low-density parity-check codes. A search on BIDS returns the following citations: [14], [25], [28], [63], [44], [49], [48], [54], [53], [55], [59], [64], [71], [72]. Of these, it seems that the only author who pursued the practical implementation of Gallager codes (and variations on them) was Tanner [63]. An independent rediscovery of Gallager's work has been made by Wiberg [69], [68]. We regret that we initially misunderstood Gallager's work: in [40], we incorrectly asserted that Gallager's codes were "bad" owing to a confusion with their duals, low-density generator matrix codes, which are bad; we also confused the decoding algorithms of Gallager and Meier and Staffelbach.

In 1963, the N^2 cost in memory for explicit storage of the generator matrix would have been unattainable, so computational resources were (temporarily) a problem. R. G. Gallager (personal communication) has suggested that Gallager codes were generally forgotten because it was assumed that concatenated codes [23] were superior for practical purposes.

H. Future Work

Generalization to q-ary alphabets. Gallager and MN codes can also be defined over q-ary alphabets consisting of the elements of GF(q). The generator matrix of a Gallager code over GF(q) takes the form $\left[\begin{smallmatrix}\boldsymbol{I}\\ -\boldsymbol{C}_{\rm n}^{-1}\boldsymbol{C}_{\rm s}\end{smallmatrix}\right]$, where the matrix $\boldsymbol{A} = [\boldsymbol{C}_{\rm s}|\boldsymbol{C}_{\rm n}]$ is a very sparse matrix with its nonzero elements drawn from the nonzero elements of GF(q). The inversion and multiplication operations are carried out in the algebra of GF(q). The decoding can be performed with a belief propagation algorithm, as with the binary Gallager codes. We are investigating the application of these codes (with $q = 4, 8, 16$) to the q-ary symmetric channel—and to the binary-symmetric channel and binary Gaussian channel, since there is no obvious reason to believe that the $q = 2$ Gallager codes are the best Gallager codes for binary channels. Our results show that Gallager codes over GF(4) and GF(8) perform better than comparable Gallager codes over GF(2) in the case of the binary-symmetric channel and the Gaussian channel [17].

Constructions. By introducing Constructions 2A and 2B, we pushed the performance of Gallager codes a little closer to capacity. Are there further useful changes we could make to the code construction? We are currently investigating the possibility of *systematic* construction of matrices $\boldsymbol{A}$ whose corresponding graphs have large girth [44], [8], [35].

In this paper we have mainly considered *regular* low-density matrices, that is, matrices in which the weight per column is constant and the weight per row is constant, or nearly constant. It is obviously a disappointment that, whereas the way to obtain very good codes is to increase the density, the sum-product algorithm performs worse for denser matrices. There is a way out of this dilemma however: we obtained better performance by using slightly irregular matrices with weight two and weight three columns (see Fig. 9); Luby, Mitzenmacher, Shokrollahi, and Spielman [36] have recently extended this idea, investigating highly irregular Gallager codes. Their results indicate that significant enhancements in performance can be obtained. We have applied this idea to Gallager codes over GF(8) and have discovered an irregular Gallager code with blocklength 24 000 bits whose performance equals that of the best turbo codes [18]. The choice of construction of Gallager code remains a productive area for further research.

Bursty channels and fading channels. Since Gallager codes are, given an optimal decoder, good codes for any channel in a wide class, we are optimistic that they will be excellent codes for channels with bursts and fades. We anticipate that the sum-product algorithm can be generalized to handle simultaneous equalization and decoding. Only if we model and infer the channel variations will we be able to get close to the Shannon limit of such time-varying channels. In contrast, many codes handle bursts by *interleaving*, that is, reordering the bits so that the bursts look like uniform noise.

Our results on MN codes serve as initial results describing the performance of Gallager codes for decoding in the presence of bursts. Consider a two-state channel which flips between a high noise state with $f_{\rm n} = 0.5$ and a low noise state with $f_{\rm n} = 0.114$. The rate $1/4$ code of blocklength 13 298 bits can communicate reliably over this channel if the burst periods are identified, as long as the fraction of time spent in the high noise state is less than 25%.

In contrast, if we used a traditional interleaving method to cope with the bursts, the effective noise level would be $\bar{f} = 0.25 \times 0.5 + 0.75 \times 0.114 = 0.21$, for which the capacity is $C \simeq 0.26$. It seems unlikely that there is a practical rate $1/4$ interleaved code that can communicate reliably under these conditions.

Cryptanalysis. This work grew out of an interest in a problem in cryptanalysis [3], the inference of the state of a linear feedback shift register given its noisy output sequence, which is also equivalent to the decoding of a cyclic code. The free energy minimization algorithm was found to be an improvement over Meier and Staffelbach's algorithm in [38]. We anticipate that the sum-product decoder might perform even better on these cryptanalysis problems. We are at present investigating this possibility.

Statistical Physics. Consider a set of L spins $\xi_l = \pm 1$ among which there are M couplings of order $t_r = 8$ such that the Hamiltonian is

$$\begin{aligned} E(\xi) = {} & \sum_{l=1}^{L} b_l \xi_l \\ & + \sum_{m=1}^{M} J_m \xi_{i_1(m)} \xi_{i_2(m)} \xi_{i_3(m)} \xi_{i_4(m)} \xi_{i_5(m)} \xi_{i_6(m)} \\ & \times \xi_{i_7(m)} \xi_{i_8(m)}. \end{aligned} \tag{62}$$

We have assumed that spins are coupled together in groups of eight in order to obtain a relationship to a matrix $\boldsymbol{A}$ with weight 8 per row. If we identify $J_m = 2z_m - 1$ and set up the functions $i_\tau(m)$ to correspond to the 1's in the matrix $\boldsymbol{A}$, then the decoding problem maps onto the task of finding the ground state of this energy function in the limit of small b_l.

We have found that the sum-product decoder is a better algorithm than the free energy minimization algorithm (Appendix C) for solving this problem. A difference between the algorithms is that the free energy minimization algorithm (also known as "mean field theory") shows spontaneous symmetry breaking, whereas the sum-product algorithm only breaks symmetry if the energy function itself breaks symmetry. Prior work related to this concept is found in [65].

It is possible that further benefits may be obtained by applying sum-product concepts in statistical physics or to other optimization problems where mean field methods have been found useful [30], [2].

Decoding algorithms. We conjecture that as we get closer to the Shannon limit, the decoding problem gets harder. But we do not understand what aspects of the problem determine the practical limits of our present decoding algorithms. It would be interesting to obtain a convergence proof for the sum-product algorithm and to develop ways of reducing the inaccuracies introduced by the approach of ignoring the cycles present in the belief network.

Appendix A
Real-Output Channels

In the main body of the paper our theorems and discussions have focussed on binary-input, binary-output channels of two types: the memoryless binary-symmetric channel and the more general symmetric stationary ergodic binary channel (Definition 6). We proved that MN codes and Gallager codes are good for such channels. The same codes are also good for channels with other output alphabets as long as they satisfy simple symmetry and law-of-large-numbers properties. We have in mind binary-input channels with real outputs such as the additive white Gaussian noise channel and fading channels.

We start by discussing the simple case of a channel with inputs of $t_{\rm n} = \pm 1$ and real-valued outputs. Here, an implicit noise vector $\boldsymbol{n}$ can be envisioned on the basis of an arbitrarily selected received vector $\boldsymbol{r}$, with $\boldsymbol{r} = 0$ being perhaps the most convenient choice, since it simplifies the computation of the syndrome $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{r} \bmod 2 = 0$ and (63) below. For independent noise, the bits of $\boldsymbol{n}$ are probabilistically independent, with the probability of a bit being 1 being determined by the likelihood ratio for the received signal. For a Gaussian channel with inputs of $x_{\rm in} = \pm x$, the received signal is $y_{\rm n} = (2t_{\rm n} - 1)x + \nu_{\rm n}$, where $\nu_{\rm n}$ is zero-mean Gaussian noise of variance σ^2. The effective probability for a 1 in bit i of the noise vector $\boldsymbol{n}$ (based on $\boldsymbol{r} = 0$) is then

$$f_{{\rm n}(i)} = \frac{1}{1 + e^{-2y_i x/\sigma^2}}. \tag{63}$$

(Note that this probability can be greater than $1/2$, in keeping with the arbitrary nature of the choice of $\boldsymbol{r} = 0$.) In any given realization of the noise we can thus deduce an effective binary noise distribution $P(\boldsymbol{n})$ for an equivalent time-varying binary-symmetric channel. Whether the decoding problem is solvable depends on the entropy of this distribution. We thus need to add to our list of error types (Section II-C) a third possibility:

III. The distribution $P(\boldsymbol{n})$ has entropy greater (by some η) than the mean entropy. Let the probability of this event be $P_{\rm III}$.

This failure mode occurs with a probability determined by the distribution of large deviations of the channel. This probability obviously must vanish with increasing blocklength for our codes to be good.

A. Extension of Proofs to Channels with Nonbinary Outputs and Temporal Correlations

Gallager and MN codes are good for channels that are stationary and ergodic, that have vanishing $P_{\rm III}$ and that satisfy this symmetry property:

Definition 8: A temporal binary-input channel is *symmetric*[1] if the optimizing distribution $P_N^*(\boldsymbol{t})$ of the channel is a uniform distribution $P_N^*(\boldsymbol{t}) = 1/2^N$.

If this symmetry property is satisfied then it is evident that the decoding problem is equivalent to the decoding of a symmetric stationary ergodic binary channel (Definition 6).

B. Gaussian Channel Definitions

We simulated the following Gaussian channel. The binary inputs are $x_{\rm in} = \pm x$ and the real output y has a conditional distribution that is Gaussian with mean $x_{\rm in}$ and variance σ^2

$$P(y \,|\, x_{\rm in}) = {\rm Normal}\,(x_{\rm in}, \sigma^2). \tag{64}$$

For convenience we set $\sigma = 1$ and varied x to change the signal-to-noise ratio. The capacity of the channel can be defined in a couple of ways. If we were allowed to use arbitrary input values with the same mean power as the binary inputs $x_{\rm in} = \pm x$, the capacity would be

$$C_{\rm Unconstrained} = C_U = \frac{1}{2} \log_2 \left(1 + \frac{x^2}{\sigma^2} \right). \tag{65}$$

[1] Our definition of a "symmetric" channel differs from that of Cover and Thomas [16]. For them, a channel is "symmetric" if the rows of $p(y \,|\, x)$ are permutations of each other and the columns are permutations of each other; a channel is "weakly symmetric" if the rows of $p(y \,|\, x)$ are permutations of each other and the column sums are equal. This definition of a symmetric channel is too restrictive, as it cannot even encompass a Gaussian channel. The definition given here may conversely be viewed as too broad.

TABLE III
GLOSSARY
The symbol "|" between two matrices denotes concatenation, for example a systematic parity check matrix might be written $\boldsymbol{H} = [\boldsymbol{P}|\boldsymbol{I}_{N-K}]$.

Symbol	Meaning	Type
N	Transmitted blocklength of a code	Integer
K	Source blocklength of a code	Integer
$\boldsymbol{G}$	Generator matrix of a code	Binary matrix
$\boldsymbol{H}$	Parity-check matrix of a code	Binary matrix
$\boldsymbol{P}$	Parity block within a systematic generator matrix or parity check matrix, e.g., $\boldsymbol{H} = [\boldsymbol{P}\vert\boldsymbol{I}_{N-K}]$	Binary matrix
$\boldsymbol{s}$	Source string, length K	Binary vector
$\boldsymbol{t}$	Transmitted string, length N. $\boldsymbol{t} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{s}$	Binary vector
$\boldsymbol{n}$	Noise vector	Binary vector
$\boldsymbol{r}$	Received string ($\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{n}$)	Binary vector
$\boldsymbol{Hr}$	Syndrome vector	Binary vector
$\boldsymbol{A}$	Very sparse matrix of dimension $M \times L$. $\boldsymbol{A} = [\boldsymbol{C}_1\vert\boldsymbol{C}_2] = [\boldsymbol{C}_{\mathrm{s}}\vert\boldsymbol{C}_{\mathrm{n}}]$	Binary matrix
$\boldsymbol{C}_{\mathrm{s}}$	Very sparse matrix of dimension $M \times (L - M)$	Binary matrix
$\boldsymbol{C}_{\mathrm{n}}$	Very sparse square matrix of dimension $M \times M$	Binary matrix
L	Number of columns in $\boldsymbol{A}$. In Gallager codes $L = N$. In MN codes $L = N + K$.	Integer
M	Number of rows in $\boldsymbol{A}$. In Gallager codes $M = N - K$. In MN codes $M = N$.	Integer
t	Number of 1's per column in $\boldsymbol{A}$ (Gallager's j)	Integer or real
t_r	Number of 1's per row in $\boldsymbol{A}$ (Gallager's k)	Integer or real
ρ	Ratio $(L - M)/M$. Symbol rate of MN code.	Real
λ	Ratio L/M. $\lambda = \rho + 1$.	Real > 0
$\boldsymbol{x}$	possibly sparse vector of length L	Binary vector
$\boldsymbol{z}$	vector of length M such that $\boldsymbol{Ax} = \boldsymbol{z} \bmod 2$	Binary vector
w	weight of vector $\boldsymbol{x} - \boldsymbol{x}'$; equivalently, the number of columns of $\boldsymbol{A}$ that might be linearly dependent	Integer
$p_{00}^{(r)}$	Probability that random walk on M-dimensional hypercube returns to starting corner on step r	Real $\in [0, 1]$
$q_{00}^{(r)}$	Upper bound for $p_{00}^{(r)}$	Real
k, l, m, n	indices running from 1 to K, L, M, N.	Integer
j	index running from 1 to $M/2$ in the eigenvalue expansion of $p_{00}^{(r)}$.	Integer
κ	j/M	Real $\in [0, 1/2]$
ϕ	w/L	Real $\in [0, 1]$
f	Density of $\boldsymbol{x}$	Real
f_{n}	Noise density	Real
f_{s}	Source density	Real
r	Number of steps in random walk on M-dimensional hypercube. $r = wt$. $r/M = \lambda\phi t$.	Integer
q_{ml}^x, r_{ml}^x	Probabilities in sum-product algorithm	Real
(n, k, t)	Traditional labels for blocklength, source blocklength, and maximum number of errors that can be corrected. N.B., t above is different.	Integers
d	Minimum distance of a code.	Integer

If we accept the constraint that only the defined binary inputs are allowed, then the capacity is reduced to

$$\begin{aligned} C_{\text{Binary}} = C_B &= H(Y) - H(Y\,|\,X) && (66)\\ &= -\int dy\; P(y)\log P(y) \\ &\quad + \int dy\; P(y\,|\,x=x_0)\log P(y\,|\,x=x_0) && (67)\end{aligned}$$

where

$$P(y) = \frac{1}{2\sqrt{2\pi\sigma^2}}\left[e^{-(y-x)^2/(2\sigma^2)} + e^{-(y+x)^2/(2\sigma^2)}\right] \quad (68)$$

which may be evaluated numerically. If one communicates over the Gaussian channel using a code of rate R then it is conventional to describe the signal-to-noise ratio by

$$\frac{E_b}{N_0} = \frac{x^2}{2R\sigma^2} \quad (69)$$

and to report this number in decibels as $10\log_{10} E_b/N_0$.

Appendix B
Reporting of Empirical Results

A. Error Bars

The experiments result in a certain number of block decoding failures r out of a number of trials n. We report the maximum-likelihood estimate of the block error probability, $\hat{p} = r/n$, and a confidence interval $[p_-, p_+]$, defined thus: if $r \geq 1$ then

$$p_\pm = \hat{p}\,\exp\left(\pm 2\sigma_{\log p}\right), \qquad \text{where } \sigma_{\log p} = \sqrt{(n-r)/(rn)};$$

else if $r = 0$, $p_+ = 1 - \exp(-2/n)$, and $p_- = 0$. When reporting the *bit* error probability we use the error bars derived from the block error probability; we do not bother including the additional uncertainty in the bit error rate within erroneous blocks, which is expected to be much smaller than the uncertainty in the block error probability.

B. Comparison with Other Codes

Performances of RM and BCH codes were computed assuming for an RM code of minimum distance d that more than $\lfloor d/2 \rfloor$ errors cannot be corrected. For BCH codes it was assumed that more than t errors cannot be corrected, 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 [9], "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." All relevant BCH codes listed in [56] are included (block sizes up to 1023).

Appendix C
Decoding by Free Energy Minimization

MacKay [38], [37] derived a continuous optimization algorithm for solving the discrete decoding problem

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{z} \bmod 2 \quad (70)$$

where $\boldsymbol{A}$ is a given binary $M \times L$ matrix and $\boldsymbol{z}$ is a received vector of length M. The vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are assumed to have a prior distribution that is separable thus

$$P(\boldsymbol{x},\boldsymbol{y}) = \prod_{l=1}^{L} P(x_l) \prod_{m=1}^{M} P(y_m).$$

The algorithm is only practically useful for matrices $\boldsymbol{A}$ that are sparse. Problems of the form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \bmod 2 \quad (71)$$

can also be solved using the free energy minimization algorithm by solving a sequence of problems of the general form (70) with the fictitious noise level of the vector $\boldsymbol{y}$ decreasing to zero.

The algorithm works by approximating the complicated posterior probability of $\boldsymbol{x}$ given $\boldsymbol{z}$ by a simpler separable distribution

$$Q(\boldsymbol{x};\theta) = \prod_{l=1}^{L} q_l(x_l;\theta_l).$$

This distribution's parameters $\theta = \{\theta_l\}$ (one parameter for each bit of $\boldsymbol{x}$) are then adjusted so as to minimize a measure of the divergence between the approximating distribution $Q(\boldsymbol{x})$ and the true distribution $P(\boldsymbol{x}\,|\,\boldsymbol{z},\boldsymbol{A})$, the variational free energy

$$F(\theta) = -\sum_{\boldsymbol{x}} Q(\boldsymbol{x};\theta)\log\frac{P(\boldsymbol{z}\,|\,\boldsymbol{x},\boldsymbol{A})P(\boldsymbol{x})}{Q(\boldsymbol{x};\theta)}. \quad (72)$$

The evaluation of this objective function and its gradient is possible in time linear in the weight of $\boldsymbol{A}$. There is also an update algorithm for each component θ_l such that F is guaranteed to decrease. This iterative procedure is obviously not the optimal decoding algorithm, but it is practical. We originally developed MN codes with this decoding algorithm in mind, so we report some experimental results. However, this decoder has been superseded by the sum-product algorithm of Section III. Sum-product decoding is less complicated because there is no need to have an annealing schedule for a temperature parameter.

A. Empirical Results: Free Energy Minimization, Ensemble 4

A value of $f = 0.05$ was selected for experiments with Gallager codes having rate $R \simeq 1/2$ and $t = 4$, using Construction 1A.

We found that as the block size N was increased the performance improved. The block error probabilities for rate $1/2$ codes with block lengths of 2000, 4000, 8000, and 20 000 were $0.11, 0.046, 0.0058$, and 0.00017. We also found that with larger and smaller values of t than $t = 4$, the code did not work as well when decoded by free energy minimization.

B. Insights into the Source of Errors

We examined some of the errors made by the free energy minimization decoder and found that they tended to occur when the vector $\boldsymbol{x}$ was such that another slightly different typical vector $\boldsymbol{x}'$ had a similar (but not identical) encoding $\boldsymbol{z}'$. In terms of the random walk on the hypercube (Section II-C), these errors correspond to walks that after a small number of steps return close to the starting corner. They were possible because of rare topologies in the network corresponding to the $\boldsymbol{A}$ matrix such as the topology illustrated in Fig. 5(c). We can eliminate the possibility of these errors by modifying the ensemble of matrices $\boldsymbol{A}$ so that the corresponding network does not have short cycles in it.

C. Empirical Results: Free Energy Minimization, Construction 1B

We made new $\boldsymbol{A}$ matrices by taking matrices from Ensemble 4 with $t = 4$ and deleting columns until there were no short loops of the type shown in Fig. 5(e). These matrices, having fewer columns, correspond to codes with slightly lower rates. They also have nonuniform weight per row, which may make them slightly suboptimal. We found that the topological modifications gave codes which were able to communicate at slightly higher rates over slightly noisier channels with a smaller probability of error. A summary of our results is that in terms of block error probability for a given communication rate, Gallager and MN codes, *when decoded by free energy minimization*, can be superior to Reed–Muller codes, and Gallager codes can outperform BCH codes by a small margin.

Significantly better results were obtained using the sum-product decoder described in the main body of this paper.

D. Contrast with Sum-Product Algorithm

We believe the reason the sum-product algorithm performs much better than the variational free energy minimization (mean field) algorithm is that the mean field algorithm exhibits spontaneous symmetry breaking. It is possible for a cluster of bits, whose state has not been determined by the influence of the data, to collapse into a locally consistent state. The sum-product algorithm (at least in the ideal case of a graph without cycles) does not show any such spontaneous symmetry breaking.

Appendix D
Inequalities

We prove the following inequalities:

$$\frac{1}{N+1}2^{NH_2(K/N)} \leq \binom{N}{K} \leq 2^{NH_2(K/N)}. \tag{73}$$

In general, if $K_1 + K_2 + \cdots K_I = N$ and $H(p_1, p_2, \cdots p_I) \equiv \sum_i p_i \log_2 \frac{1}{p_i}$ then

$$\frac{N!}{K_1! \cdots K_I!} \leq 2^{NH(K_1/N, K_2/N, \cdots, K_I/N)}. \tag{74}$$

Proofs of Inequalities (73) and (74): Proof of right-hand inequality: consider the multinomial distribution

$$P(K_1, K_2, \cdots K_I | p_1, p_2, \cdots p_I) = \frac{N!}{K_1! \cdots K_I!} \prod_i p_i^{K_i}. \tag{75}$$

Set $p_i = K_i/N$. Then evaluate the probability, which we know is less than 1

$$\frac{N!}{K_1! \cdots K_I!} \prod_i 2^{K_i \log_2(K_i/N)} \leq 1. \tag{76}$$

$$\rightarrow \frac{N!}{K_1! \cdots K_I!} \leq 2^{\left(-N \sum_i K_i/N \log_2(K_i/N)\right)}. \tag{77}$$

The left-hand inequality in (73) is proved by considering again the quantity

$$P(k) = \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k}. \tag{78}$$

If $p = K/N$ then $P(K)$ is greater than or equal to $P(k)$ for all k. So

$$(N+1)P(K) \geq \sum_k P(k) = 1 \tag{79}$$

$$\begin{aligned} \rightarrow \binom{N}{K} &\geq \frac{1}{N+1}(K/N)^{-K}(1-K/N)^{-(N-K)} \\ &= \frac{1}{N+1}2^{NH_2(K/N)}. \end{aligned} \tag{80}$$

Similarly, we may prove that, for $K < N/2$

$$\sum_{k=0}^{K} \binom{N}{k} \leq 2^{NH_2(K/N)}. \tag{81}$$

Proof: Consider the sum of binomial terms

$$\sum_{k=0}^{K} P(k) = \sum_{k=0}^{K} \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k} \leq 1 \tag{82}$$

Setting $p = K/N$, we examine the factor $(1-p)/p = (N-K)/K$. Because $K < N/2$, this factor is greater than 1. So, dividing each term in the sum by $(1-p)/p$ an appropriate number of times

$$\sum_{k=0}^{K} \frac{N!}{k!(N-k)!} p^K (1-p)^{N-K} \leq 1 \tag{83}$$

$$\rightarrow \sum_{k=0}^{K} \binom{N}{k} \leq 2^{(-K \log_2(K/N) - (N-K)\log_2((N-K)/N))}. \tag{84}$$

We also note the following inequalities:

$$x! \geq x^x e^{-x} e \quad \text{or equivalently} \quad \log x! \geq x \log x - x + 1 \tag{85}$$

The proof is straightforward by integration of $\log x$. □

Appendix E
Bounds on Random Walk's Return Probability

We derive several upper bounds on the probability $p_{00}^{(r)}$ that the random walk on the M-dimensional hypercube returns to its starting corner on the rth step. We use the result from [31] that

$$p_{00}^{(r)} = 2^{-M}\sum_{j=0}^{M}\binom{M}{j}\left(1-\frac{2j}{M}\right)^{r}. \tag{86}$$

See [13], [20], [32], [34], and [62] for further information about this random walk, which is also known as the Ehrenfest Urn model. Equation (86) is an eigenvalue expansion, where the eigenvalues of the Markov process are labeled by $j = 0\cdots M$ and have value $(1-\frac{2j}{M})$. For every positive eigenvalue there is an opposite negative eigenvalue. We note that for odd r, $p_{00}^{(r)}$ is zero, and that for even r, $p_{00}^{(r)}$ is a decreasing function of r. For convenience we can obtain a monotonic upper bound on $p_{00}^{(r)}$ by restricting the sum over j to the positive eigenvalues and including a leading factor of 2.

$$p_{00}^{(r)} \le q_{00}^{(r)} \equiv 2\times 2^{-M}\sum_{j=0}^{M/2}\binom{M}{j}\left(1-\frac{2j}{M}\right)^{r}. \tag{87}$$

We now derive various bounds valid for all $r > 0$; some bounds are tight for $r \ll M$ and some for $r \gg M$. We also evaluate a numerical bound that is useful for all r.

A. Bound that is Tight for $r \gg M$

$$\begin{aligned} q_{00}^{(r)}/2 &= 2^{-M}\sum_{j=0}^{M/2-1}\binom{M}{j}\left(1-\frac{2j}{M}\right)^{r} && (88)\\ &\le 2^{-M}\sum_{j=0}^{M/2-1}\frac{M^j}{j!}e^{-r\frac{2j}{M}} && (89)\\ &\le 2^{-M}\sum_{j=0}^{\infty}\frac{1}{j!}e^{-(\frac{2r}{M}-\log M)j}. && (90)\end{aligned}$$

At (89) we have used the inequality $(1-a)^r \le e^{-ar}$. We note that

$$\sum_{j=0}^{\infty}\frac{1}{j!}e^{-aj} = e^{(e^{-a})} \tag{91}$$

and obtain

$$\begin{aligned} q_{00}^{(r)}/2 &\le 2^{-M}\exp\left[\exp\left(\frac{2r}{M}-\log M\right)\right] && (92)\\ &= 2^{-M}\exp\left[Me^{-2r/M}\right] && (93)\\ \Rightarrow q_{00}^{(r)} \le q_b^{(r)} &\equiv 2\exp\left[M(e^{-2r/M}-\log 2)\right]. && (94)\end{aligned}$$

The logarithm of this bound is

$$\log\left(q_b^{(r)}/2\right) = M(e^{-2r/M}-\log 2). \tag{95}$$

B. Bound that is Tight for $r \ll M$

We consider the logarithm of one term in the sum over j in (87). We obtain a bound on the maximum value over j that this quantity can assume. We define $\kappa = j/M$ and $\mu = 1-2\kappa$.

$$\begin{aligned} \log\left[\binom{M}{j}\left(1-\frac{2j}{M}\right)^{r}\right] &\le MH_2^e(\kappa) + r\log(1-2\kappa) && (96)\\ &\le M\left(\log 2 - \frac{1}{2}\mu^2\right) + r\log(\mu). && (97)\end{aligned}$$

At (96) we use inequality (73) and at (97) we use the inequality $H_2^e(\kappa) \le \log 2 - \frac{1}{2}(1-2\kappa)^2$. We differentiate and find that the maximum of this function is at $\mu^2 = r/M$, so that we can bound every term as follows for any j:

$$\log\left[2^{-M}\binom{M}{j}\left(1-\frac{2j}{M}\right)^{r}\right] \le -\frac{1}{2}r + \frac{1}{2}r\log\frac{r}{M}. \tag{98}$$

We thus conclude that for any $r > 0$

$$\begin{aligned} q_{00}^{(r)} &\le 2\times 2^{-M}\sum_{j=0}^{M/2-1}\binom{M}{j}\left(1-\frac{2j}{M}\right)^{r} && (99)\\ &\le \frac{M}{2}2\exp\left[-\frac{1}{2}r + \frac{1}{2}r\log\frac{r}{M}\right] && (100)\\ \Rightarrow q_{00}^{(r)} \le q_c^{(r)} &\equiv Me^{-r/2}\left(\frac{r}{M}\right)^{r/2}. && (101)\end{aligned}$$

This bound illustrates the important power law decrease of $q_{00}^{(r)}$ for small r. The logarithm of this bound is

$$\begin{aligned} \log q_c^{(r)} &= \log M - \frac{r}{2} + \frac{1}{2}r\log\frac{r}{M} && (102)\\ &= \log M + \frac{1}{2}M\left(-\frac{r}{M} + \frac{r}{M}\log\left(\frac{r}{M}\right)\right). && (103)\end{aligned}$$

The leading factor of M in (101) is the undesirable side-effect of two of the inequalities.

C. Bound That Is a Factor of M Tighter for $r \ll M$

The above bound is tight enough to prove the main theorems of the paper for $t \ge 4$ but not for $t = 3$, because of the leading factor of M, so we now improve on it.

We count the number of walks of length r (where r is even) that return to the origin. A walk that returns to the origin can be created as follows. First we select (with replacement) $r/2$ directions from the M axes of the hypercube as the directions that will be traversed twice. The number of different selections is equal to the number of ways of putting $r/2$ indistinguishable objects in M bins, which is the number of distinct ways of ordering $M-1$ indistinguishable partitions and $r/2$ indistinguishable objects, which is $\binom{M-1+r/2}{r/2}$. Then having decided which are the directions we will traverse, the number of distinct walks that can be created is less than or equal to the number of distinct orderings of $r/2$ pairs of indistinguishable objects, $\frac{r!}{2^{r/2}}$, with equality in the case where all directions are traversed twice or zero times. Thus the total number of walks of length r that return to the origin, $n_{00}^{(r)}$, is bounded by

$$n_{00}^{(r)} \le \binom{M-1+r/2}{r/2}\frac{r!}{2^{r/2}} \tag{104}$$

and so, since the total number of walks of length r is M^r

$$p_{00}^{(r)} = \frac{n_{00}^{(r)}}{M^r} \le \frac{1}{M^r}\binom{M-1+r/2}{r/2}\frac{r!}{2^{r/2}}. \tag{105}$$

We loosen this bound into a slightly more convenient form, introducing a largest value of r, $r^* = \gamma M$, for which we plan to use this bound. We will set γ at our convenience to a value independent of M.

$$p_{00}^{(r)} \le \frac{1}{M^r}\frac{(M-1+\frac{r^*}{2})^{r/2}}{(r/2)!}\frac{r!}{2^{r/2}}, \qquad \text{for } r \le r^* = \gamma M \tag{106}$$

$$\Rightarrow p_{00}^{(r)} \le (1/2+\gamma/4)^{r/2}\left(\frac{r}{M}\right)^{r/2}, \qquad \text{for } r \le r^* = \gamma M. \tag{107}$$

For our proof we set $\gamma = 1$ so that

$$p_{00}^{(r)} \le \left(\frac{3}{4}\frac{r}{M}\right)^{r/2}, \qquad \text{for } r \le M. \tag{108}$$

This bound decreases monotonically in r up to $r' \equiv \gamma' M$, where $\gamma' = \frac{4}{3e}$ (found by differentiation), so, since we know $p_{00}^{(r)}$ is bounded by a decreasing function, we obtain

$$p_{00}^{(r)} \le q_d^{(r)} \equiv \begin{cases} \left(\frac{3}{4}\frac{r}{M}\right)^{r/2}, & r \le r' = \gamma' M \left(\text{where } \gamma' = \frac{4}{3e}\right) \\ \left(\frac{3}{4}\gamma'\right)^{r'/2}, & r > r'. \end{cases} \tag{109}$$

The logarithm of $q_d^{(r)}$ is

$$\log q_d^{(r)} = \frac{r}{2}\log\left(\frac{3}{4}\frac{r}{M}\right), \qquad \text{for } r \le r' = \gamma' M. \tag{110}$$

D. Numerical Bound Useful for All r

We continue from (96) and derive a numerical bound

$$\log\left[\binom{M}{j}\left(1-\frac{2j}{M}\right)^r\right] \le MH_2^e(\kappa) + r\log(1-2\kappa). \tag{111}$$

We differentiate with respect to κ to find the maximum value of this quantity; then we can bound the sum $q_{00}^{(r)}$ by the number of terms in the sum times the maximum value of the summand. Thus

$$q_{00}^{(r)} = 2 \times 2^{-M}\sum_{j=0}^{M/2-1}\binom{M}{j}\left(1-\frac{2j}{M}\right)^r \tag{112}$$

$$\le M\exp[-M\log 2 + MH_2^e(\kappa^*) + r\log(1-2\kappa^*)] \tag{113}$$

or, equivalently,

$$\log q_{00}^{(r)} \le \log M + M\left[-\log 2 + H_2^e(\kappa^*) + \frac{r}{M}\log(1-2\kappa^*)\right] \tag{114}$$

where κ^* is the solution of

$$\log\left(\frac{1-\kappa}{\kappa}\right) - \frac{2r/M}{(1-2\kappa)} = 0 \tag{115}$$

or, equivalently,

$$\kappa = \frac{1}{1+\exp\left[\frac{2r/M}{(1-2\kappa)}\right]}. \tag{116}$$

We can solve this equation numerically by iteration, starting (for example) from $\kappa = 0$, and setting κ equal to κ', the value of the right-hand side, repeating until convergence is established. Convergence, which can be slow, may be accelerated by setting κ equal to $(\kappa+\kappa')/2$.

For large M and $r \gg 1$, the $\log M$ term is of small order and we neglect it in our numerical computations, using

$$\log q_{00}^{(r)} \lesssim \log q_e^{(r)} \equiv M\left[-\log 2 + H_2^e(\kappa^*) + \frac{r}{M}\log(1-2\kappa^*)\right]. \tag{117}$$

E. Another Good Bound for Large r

Using the same method as in Appendix E-D, we can obtain an explicit bound

$$\log\left[\binom{M}{j}\left(1-\frac{2j}{M}\right)^r\right] \le MH_2^e(\kappa) + r\ \log(1-2\kappa) \tag{118}$$

Now H_2 is a convex function, so it is upper-bounded by its tangent. For any choice of κ^*, and for all κ

$$H_2^e(\kappa) \le H_2^e(\kappa^*) + (\kappa-\kappa^*)\log\left(\frac{(1-\kappa^*)}{\kappa^*}\right). \tag{119}$$

So

$$\log\left[\binom{M}{j}\left(1-\frac{2j}{M}\right)^r\right] \le MH_2^e(\kappa^*) + (\kappa-\kappa^*)\log\left(\frac{(1-\kappa^*)}{\kappa^*}\right) + r\log(1-2\kappa). \tag{120}$$

We differentiate with respect to κ to find the maximum value of this quantity, which is at

$$\kappa = \frac{1}{2} - \frac{2}{M\log\left(\frac{(1-\kappa^*)}{\kappa^*}\right)} \tag{121}$$

then we can bound the sum $q_{00}^{(r)}$ by the number of terms in the sum times the maximum value of the summand, choosing κ^* to have any convenient value. Based on what we have learned from the previous bound, we choose

$$\kappa^*(r) = \frac{1}{1+\exp\left(\frac{2r}{M}\right)} \tag{122}$$

and obtain (after straightforward algebra)

$$q_{00}^{(r)} \le q_f^{(r)} \equiv M\exp\left[-M\log 2 + MH_2^e\left(\kappa^*(r)\right) - 2r\kappa^*(r)\right]. \tag{123}$$

The five bounds $q_b^{(r)}$, $q_c^{(r)}$, $q_d^{(r)}$, $q_e^{(r)}$, and $q_f^{(r)}$ are plotted in the large M limit in Fig. 22.

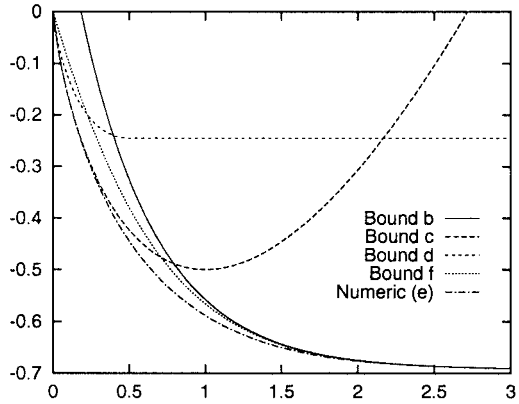


Fig. 22. Plot of five bounds on the function $\frac{1}{M}\log p_{00}^{(r)}$, for large M, as a function of r/M.
Bound b: $\frac{1}{M}\log q_b^{(r)} = (e^{-2r/M} - \log 2)$.
Bound c: $\frac{1}{M}\log q_c^{(r)} = \frac{1}{2}(-r/M + r/M\log(r/M))$.
Bound d: $q_d^{(r)} = (\frac{3}{4}\frac{r}{M})^{r/2}$ for $r \leq r' = \gamma' M$, where $\gamma' = \frac{4}{3e}$.
Bound f: $q_f^{(r)} = M\exp[-M\log 2 + MH_2^e(\kappa^*(r)) - 2r\kappa^*(r)]$ where $\kappa^*(r) = \frac{1}{1+\exp(\frac{2r}{M})}$.
Numerical bound $\frac{1}{M}\log q_e^{(r)}$ computed as described in Appendix E-D. .

Appendix F
Evaluation of Upper and Lower Bounds on Achievable Information Rate in Case of Noise of Uniform Density

We start from (21). In the case of uniform density, $h(w\,|\,\boldsymbol{x})$ depends only on the weight of $\boldsymbol{x}$, which we will denote $u_{\boldsymbol{x}}$.

$$\bar{P}_{\rm II} \leq \sum_{w=1}^{L}\sum_{\boldsymbol{x}\in T} P(\boldsymbol{x})h(w\,|\,u_{\boldsymbol{x}})q_{00}^{(wt)}. \tag{124}$$

We obtain a numerical lower bound on the achievable rate of Gallager codes by computing as a function of w a bound on $g(w) = \sum_{\boldsymbol{x}\in T} h(w\,|\,u_{\boldsymbol{x}})$, the number of pairs $(\boldsymbol{x}, \boldsymbol{x}')$ that give rise to an $(\boldsymbol{x} - \boldsymbol{x}')$ of weight w.

First consider the number of pairs such that $\boldsymbol{x}$ has weight u, $\boldsymbol{x}'$ has weight u', and $(\boldsymbol{x} - \boldsymbol{x}')$ has weight w. This quantity $g(u, u', w)$ can be written in terms of $\Delta = (u - u')/2$ and $\bar{l} = (u + u')/2$ as

$$g(u, u', w) = \frac{L!}{(\frac{w}{2}+\Delta)!(\frac{w}{2}-\Delta)!(\bar{l}-\frac{w}{2})!(L-\bar{l}-\frac{w}{2})!} \tag{125}$$

where it is understood that $g = 0$ whenever any of the terms $v!$ in the denominator has $v < 0$. We arrive at this expression by considering the number of ways of subdividing L bits into four blocks: the 1 bits found in $\boldsymbol{x}$ but not $\boldsymbol{x}'$; the 1 bits found in $\boldsymbol{x}'$ but not $\boldsymbol{x}$; the 1 bits common to $\boldsymbol{x}$ and $\boldsymbol{x}'$; and the 0 bits common to $\boldsymbol{x}$ and $\boldsymbol{x}'$.

The number of pairs $g(w) = \sum_{u,u'} g(u, u', w)$ is bounded by the number of terms in the sum times the maximum value

$$g(w) \leq (2\eta L)^2 \max_{u,u'} g(u, u', w). \tag{126}$$

We can bound the maximum by minimizing the products $(\frac{w}{2}+\Delta)!(\frac{w}{2}-\Delta)!$ and $(\bar{l}-\frac{w}{2})!(L-\bar{l}-\frac{w}{2})!$ in the denominator. The first product is minimized by setting $\Delta = 0$; the second by setting $\bar{l}$ to its largest value (assuming that the density $f + \eta < 1/2$). Thus

$$\begin{aligned} g(w) &\leq g^*(w) \\ &\equiv (2\eta L)^2 \frac{L!}{(\frac{w}{2})!(\frac{w}{2})!(f^+L - \frac{w}{2})!((1-f^+)L - \frac{w}{2})!} \end{aligned} \tag{127}$$

where $f^+ = f + \eta$.

The function $g^*(w)$ is an increasing function of w for $0 \leq w \leq 2f(1-f)L$, where it has a sharp maximum. It is well approximated by $\log g^*(w) \simeq L\tilde{G}(\phi)$, where $\phi = w/L$ and

$$\begin{aligned} \tilde{G}(\phi) \equiv &-\phi\log(\phi/2) + (f - \phi/2)\log(f - \phi/2) \\ &+ (1 - f - \phi/2)\log(1 - f - \phi/2). \end{aligned} \tag{128}$$

The step to (128) made use of inequality (74). We find achievable rates by finding values of f and λ such that the maximum value over ϕ and κ of $\tilde{G}(\phi) + \frac{1}{\lambda M}\log q_e^{(\phi t\lambda M)}$ is just less than zero (see Appendix E-D).

Appendix G
Arithmetic Coding for Creation of Sparse Sources

A redundant source having density less than 0.5, with consecutive source symbols that are independent and identically distributed, can be produced from a dense source by using a variation on arithmetic coding [70], [57]; one simply reverses the role of encoder and decoder in a standard arithmetic coder based on a model corresponding to the sparse messages. The following pseudocode gives an algorithm for this task, but ignores issues of initialization and termination.

Loop to read a dense stream and output a sparse stream with density f.

```
loop
   At this point, $0 < R \leq 2^{k-1}$ and $0 \leq V < R$.
   while $R \leq 2^{k-2}$
      $R \leftarrow 2R$
      $V \leftarrow 2V +$ next input bit
   end while
   $M \leftarrow \lfloor fR \rfloor$
   if $V < M$ then
      output "1" bit
      $R \leftarrow M$
   else
      output "0" bit
      $V \leftarrow V - M$
      $R \leftarrow R - M$
   end if
end loop
```

Loop to reconstruct the original dense stream from its encoding as a sparse stream.

```
loop
   At this point, $0 < R \leq 2^{k-1}$ and $0 < V + R \leq 2^k$.
   while $R \leq 2^{k-2}$
      if $2^{k-2} \leq V < 2^{k-1}$ then
         $V \leftarrow V - 2^{k-2}$
         $d \leftarrow d + 1$
      else
      if $V \geq 2^{k-1}$ then
        $V \leftarrow V - 2^{k-1}$
        output a “1” bit followed by $d$ “0” bits
      else
        output a “0” bit followed by $d$ “1” bits
      endif
      $d \leftarrow 0$
    end if
    $R \leftarrow 2R$
    $V \leftarrow 2V$
  end while
  $M \leftarrow \lfloor fR \rfloor$
  if next input bit $= 1$ then
    $R \leftarrow M$
  else
    $R \leftarrow R - M$
    $V \leftarrow V + M$
  end if
end loop
```

Acknowledgment

Radford Neal made immense contributions to this work, including writing Appendix G and giving detailed input throughout; collaborating with him is a privilege.

I thank Roger Sewell, Robert McEliece, Dave Forney, and David Aldous for helpful discussions and M. D. MacLeod and the Computer Laboratory, Cambridge University for kind loans of books. I thank Geoff Hinton for generously supporting my visits to the University of Toronto.

References

[1] R. Ahlswede and G. Dueck, “Good codes can be produced by a few permutations,” *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 430–443, May 1982.

[2] S. V. B. Aiyer, M. Niranjan, and F. Fallside, “A theoretical investigation into the performance of the Hopfield model,” *IEEE Trans. Neural Networks*, vol. 1, no. 2, pp. 204–215, 1990.

[3] R. J. Anderson, “Searching for the optimum correlation attack,” in *Fast Software Encryption—Proceedings of 1994 K.U. Leuven Workshop on Cryptographic Algorithms*, B. Preneel, Ed., *Lecture Notes in Computer Science*. Springer-Verlag, 1995, pp. 179–195.

[4] S. Andreassen, M. Woldbye, B. Falck, and S. Andersen, “MUNIN—A causal probabilistic network for the interpretation of electromyographic findings,” in *Proc. 10th Nat. Conf. on AI* (AAAI: Menlo Park CA., 1987), pp. 121–123.

[5] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, 1974.

[6] G. Battail, “We can think of good codes, and even decode them,” in *Eurocode’92 26–30 October*, P. Camion, P. Charpin, and S. Harari, Eds. (Udine, Italy), in *CISM Courses and Lectures*, no 339. Wien, Austria: Springer, 1993, pp. 353–368.

[7] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite-state Markov chains,” *Ann. Math. Stat.*, vol. 37, pp. 1559–1563, 1966.

[8] F. Bein, “Construction of telephone networks,” *Notices Amer. Math. Soc.*, vol. 36, Jan. 1989.

[9] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[10] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, “On the intractability of certain coding problems,” *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 384–386, May 1978.

[11] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: Turbo-codes,” *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

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

[13] N. H. Bingham, “Fluctuation theory for the Ehrenfest urn,” *Adv. Appl. Probab.*, vol. 23, pp. 598–611, 1991.

[14] V. Chepyzhov and B. Smeets, “On a fast correlation attack on certain stream ciphers,” in *Lecture Notes in Computer Science*, vol. 547. New York: Springer-Verlag, 1991, pp. 176–185.

[15] J. T. Coffey and R. M. Goodman, “Any code of which we cannot think is good,” *IEEE Trans. Inform. Theory*, vol. 36, pp. 1453–1461, Nov. 1990.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[17] M. C. Davey and D. J. C. MacKay, “Low density parity check codes over GF(q),” *IEEE Commun. Lett.*, June 1998.

[18] ———, “Low density parity check codes over GF(q),” in *Proc. 1998 IEEE Information Theory Workshop*, June 1998.

[19] P. Delsarte and P. Piret, “Algebraic constructions of Shannon codes for regular channels,” *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 593–599, July 1982.

[20] P. Diaconis, R. L. Graham, and J. A. Morrison, “Asymptotic analysis of a random walk on a hypercube with many dimensions,” *Random Struct. Algor.*, vol. 1, pp. 51–72, 1990.

[21] D. Divsilar and F. Pollara, “On the design of turbo codes,” Tech. Rep. TDA 42–123, Jet Propulsion Lab., Pasadena, CA, Nov. 1995.

[22] G.-L. Feng and T. R. N. Rao, “Decoding algebraic-geometric codes up to the designed minimum distance,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 37–45, Jan. 1993.

[23] G. D. Forney Jr., *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.

[24] B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*. Cambridge MA: MIT Press, 1998.

[25] P. Gacs, “Reliable computation with cellular automata,” *J. Comp. Syst. Sci.*, vol. 32, no. 1, pp. 15–78, 1986.

[26] R. G. Gallager, “Low density parity check codes,” *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[27] ———, *Low Density Parity Check Codes*, no. 21 in Research Monograph Series. Cambridge, MA: MIT Press, 1963.

[28] ———, “Finding parity in a simple broadcast network,” *IEEE Trans. Inform. Theory*, vol. 34, pp. 176–180, Mar. 1988.

[29] S. W. Golomb, R. E. Peile, and R. A. Scholtz, *Basic Concepts in Information Theory and Coding: The Adventures of Secret Agent 00111*. New York: Plenum 1994.

[30] J. J. Hopfield and D. W. Tank, “Neural computation of decisions in optimization problems,” *Biol. Cybern.*, vol. 52, pp. 1–25, 1985.

[31] M. Kac, “Random walk and the theory of Brownian motion,” *Amer. Math. Monthly*, vol. 54, pp. 369–391, 1947.

[32] S. Karlin, B. Lindqvist, and Y.-C. Yao, “Markov chains on hypercubes: Spectral representations and several majorization relations,” *Random Structures and Algorithms*, vol. 4, pp. 1–36, 1993.

[33] S. L. Lauritzen and D. J. Spiegelhalter, “Local computations with probabilities on graphical structures and their application to expert systems,” *J. Roy. Statist. Soc. B*, vol. 50, pp. 157–224, 1988.

[34] G. Letac and L. Takacs, “Random walk on the m-dimensional cube,” *J. Reine Angew. Math.*, vol. 310, pp. 187–195, 1979.

[35] A. Lubotsky, “Ramanujan graphs,” *Combinatorica*, vol. 8, 1988.

[36] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Improved low-density parity-check codes using irregular graphs and belief propagation,” presented at the 1998 International Symposium on Information Theory, ISIT98, Cambridge, MA, Aug. 16–21, 1998. Also in *Proc. Int. Symp. Information Theory*, 1998, p. 117.

[37] D. J. C. MacKay, “Free energy minimization algorithm for decoding and cryptanalysis,” *Electron. Lett.*, vol. 31, no. 6, pp. 446–447, 1995.

[38] D. J. C. MacKay, “A free energy minimization framework for inference problems in modulo 2 arithmetic,” in *Fast Software Encryption—Proc. 1994 K.U. Leuven Workshop on Cryptographic Algorithms*, B. Preneel, Ed., *Lecture Notes in Computer Science*, no. 1008. New York: Springer-Verlag, 1995, pp. 179–195.

[39] ———, “Iterative probabilistic decoding of low density parity check codes,” Animations. [Online]. Available WWW: http://wol.ra.phy.cam.ac.uk/mackay/codes/gifs/, 1997..

[40] D. J. C. MacKay and R. M. Neal, “Good codes based on very sparse matrices,” in *Cryptography and Coding 5th IMA Conf.*, C. Boyd, Ed., *Lecture Notes in Computer Science*, no. 1025. Berlin, Germany: Springer, 1995, pp. 100–111.

[41] ———, “Near Shannon limit performance of low density parity check codes,” *Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996. Reprinted *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

[42] D. J. C. MacKay, S. T. Wilson, and M. C. Davey, "Comparison of constructions of irregular Gallager codes," in *Proc. 36th Allerton Conf. Commun. Contr., and Comput.,* Sept. 1998.

[43] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.

[44] G. A. Margulis, "Explicit constructions of graphs without short cycles and low-density codes," *Combinatorica*, vol. 2, no. 1, pp. 71–78, 1982.

[45] R. J. McEliece, *The Theory of Information and Coding: A Mathematical Framework for Communication*. Reading, MA: Addison-Wesley, 1977.

[46] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 140–152, 1998.

[47] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *J. Cryptol.*, vol. 1, pp. 159–176, 1989.

[48] M. J. Mihaljević and J. D. Golić, "A comparison of cryptanalytic principles based on iterative error-correction," in *Lecture Notes in Computer Science*, vol. 547. New York: Springer-Verlag, 1991, pp. 527–531.

[49] ——, "A fast iterative algorithm for a shift register initial state reconstruction given the noisy output sequence," in *Advances in Cryptology—AUSCRYPT'90*, vol. 453. New York: Springer-Verlag, 1992, pp. 165–175.

[50] ——, "Convergence of a Bayesian iterative error-correction procedure on a noisy shift register sequence," in *Advances in Cryptology—EUROCRYPT'92*, vol. 658.. New York: Springer-Verlag, 1993, pp. 124–137.

[51] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[52] W. W. Peterson and E. J. Weldon Jr.," *Error-Correcting Codes*, 2nd ed. Cambridge, MA: MIT Press, 1972.

[53] N. Pippenger, "The expected capacity of concentrators," *SIAM J. Discr. Math.*, vol. 4, no. 1, pp. 121–129, 1991.

[54] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis, "On a lower bound for the redundancy of reliable networks with noisy gates," *IEEE Trans. Inform. Theory*, vol. 37, pp. 639–643, May 1991.

[55] B. Radosavljevic, E. Arikan, and B. Hajek, "Sequential-decoding of low-density parity-check codes by adaptive reordering of parity checks," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1833–1839, Sept. 1992.

[56] T. R. N. Rao and E. Fujiwara, *Error-Control Coding for Computer Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[57] J. Rissanen and G. G. Langdon, "Arithmetic coding," *IBM J. Res. Devel.*, vol. 23, pp. 149–162, 1979.

[58] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[59] A. M. Slinko, "Design of experiments to detect nonnegligible variables in a linear model," *Cybernetics*, vol. 27, no. 3, pp. 433–442, 1991.

[60] D. A. Spielman, "Linear-time encodable and decodable error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1723–1731, Nov. 1996.

[61] L. Swanson, "A new code for Galileo," in *Proc. 1988 IEEE Int. Symp. Information Theory*, 1988, pp. 94–95.

[62] L. Takacs, "On an urn problem of Paul and Tatiana Ehrenfest," *Math. Proc. Camb. Phil. Soc.*, vol. 86, pp. 127–130, 1979.

[63] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.

[64] C. Thommesen, "Error-correcting capabilities of concatenated codes with MDS outer codes on memoryless channels with maximum-likelihood decoding," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 632–640, Sept. 1987.

[65] D. J. Thouless, P. W. Anderson, and R. G. Palmer, "Solutions of 'solvable models of a spin glass'," *Philosophical Mag.*, vol. 35, no. 3, pp. 593–601, 1977.

[66] M. A. Tsfasman, "Algebraic-geometric codes and asymptotic problems," *Discr. Appl. Math.*, vol. 33, nos. 1–3, pp. 241–256, 1991.

[67] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, 1967.

[68] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Dept. Elec. Eng., Linköping Univ., Linköping, Sweden (Linköping Studies in Science and Technology, Dissertation 440), 1996.

[69] N. Wiberg, H.-A. Loeliger, and R. Kötter, "Codes and iterative decoding on general graphs," *Euro. Trans. Telecommun.*, vol. 6, pp. 513–525, 1995.

[70] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. Assoc. Comp. Mach.*, vol. 30, no. 6, pp. 520–540, 1987.

[71] G. Zemor and G. D. Cohen, "The threshold probability of a code," *IEEE Trans. Inform. Theory*, vol. 41, pp. 469–477, Mar. 1995.

[72] M. Zivkovic, "On two probabilistic decoding algorithms for binary linear codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1707–1716, Nov. 1991.

[73] V. V. Zyablov and M. S. Pinsker, "Estimation of the error-correction complexity for Gallager low-density codes," *Probl. Pered. Inform.*, vol. 11, no. 1, pp. 23–36, 1975.