Finite-size effects and error-free communication in Gaussian channels

2000 J. Phys. A: Math. Gen. 33 1675

(http://iopscience.iop.org/0305-4470/33/8/311)

J. Phys. A: Math. Gen. **33** (2000) 1675–1681. Printed in the UK PII: S0305-4470(00)06405-2

Finite-size effects and error-free communication in Gaussian channels

Ido Kanter† and David Saad‡

† Minerva Center and Department of Physics, Bar-Ilan University, Ramat-Gan 52900, Israel
‡ The Neural Computing Research Group, Aston University, Birmingham B4 7ET, UK

Received 28 July 1999

Abstract. The efficacy of a specially constructed Gallager-type error-correcting code to communication in a Gaussian channel is examined. The construction is based on the introduction of complex matrices, used in both encoding and decoding, which comprise sub-matrices of cascading connection values. The finite-size effects are estimated for comparing the results with the bounds set by Shannon. The critical noise level achieved for certain code rates and infinitely large systems nearly saturates the bounds set by Shannon even when the connectivity used is low.

Information transmission is typically corrupted by some characteristic channel noise. Various strategies have been adopted for reducing or eliminating the noise in the received message. One of the main approaches is the use of error-correcting codes whereby the original message is encoded prior to transmission in a manner that enables the retrieval of the original message from the corrupted codeword. The maximal transmission rate is bounded by the channel capacity derived by Shannon [1] in his ground-breaking work of 1948, which does not provide specific constructions of optimal codes.

Various types of error-correcting codes have been devised over the years (for a review see [2]) for improving the transmission efficiency, most of them are generally still below Shannon's limit. We will concentrate here on a member of the parity-check codes family introduced by Gallager [3], termed the MN code [4], and on a specific construction suggested by us previously [5] for the binary symmetric channel (BSC).

The connection between parity-check codes and statistical physics was first pointed out in [6], by mapping the decoding problem onto that of a particular Ising system with multi-spin interactions. The corresponding Hamiltonian has been investigated in both fully connected [6] and diluted systems [7, 8] for deriving the typical performance of these codes; more complex architectures, somewhat similar to those examined below have been investigated in [9], establishing the connection between statistical physics and Gallager-type codes. Most of these studies have been carried out for a particular channel model, the BSC, whereby a fraction of the transmitted vector bits is flipped at random during transmission.

However, different noise models may be considered for simulating communication in various media. One of the most commonly used noise models, which is arguably the most suitable one for a wide range of applications, is that of additive Gaussian noise (usually termed additive white Gaussian noise (AWGN) in the literature). In this scenario, a message comprising N binary bits is transmitted through a noisy communication channel; a certain power level is used in transmitting the information which we will choose to be ± 1 for simplicity.

0305-4470/00/081675+07$30.00

The transmitted message is then corrupted by additive Gaussian noise of zero mean and some variance σ^2; the received (real valued) message is then decoded to retrieve the original message.

The receiver can correct the flipped bits only if the source transmits $M > N$ bits; the ratio between the original number of bits and those of the transmitted message $R \equiv N/M$ constitutes the code rate for unbiased messages. The channel capacity in the case of real-valued transmissions corrupted by Gaussian noise, which provides the bound on the maximal code rate R_c, is given explicitly [10] by

$$R_c = \tfrac{1}{2}\log(1 + v^2/\sigma^2) \tag{1}$$

where v^2 is the power used for transmission (which we take here to be ± 1) and v^2/σ^2 is therefore the signal-to-noise ratio. However, we will focus here on *binary* source messages; this reduces the maximal code rate to [10]

$$R_c = -\int \mathrm{d}y\, P(y)\log P(y) + \int \mathrm{d}y\, P(y|x = x_0)\log P(y|x = x_0) \tag{2}$$

where x is a transmitted bit (of value $x_0 = \pm 1$) and y the received bit after corruption by an additive Gaussian noise, such that

$$P(y) = \frac{1}{2\sqrt{2\pi\sigma^2}}[\mathrm{e}^{-(y-x)^2/(2\sigma^2)} + \mathrm{e}^{-(y+x)^2/(2\sigma^2)}].$$

The specific error-correcting code that we will use here is a variation of the Gallager code [3]. It became popular recently due to the excellent performance of its regular [4], irregular [11–13] and cascading connection [5] versions. In the original method, the transmitted message comprises the original message itself and additional bits, each of which is derived from the parity of a sum of certain message-vector bits. The choice of the message-vector elements used for generating single codeword bits is carried out according to a predetermined random set-up and may be represented by a product of a randomly generated sparse matrix and the message vector in a manner explained below. Decoding the received message relies on iterative probabilistic methods such as belief propagation [4, 14] or belief revision [15].

In the MN code one constructs two sparse matrices A and B of dimensionalities $M \times N$ and $M \times M$ respectively. The matrix A has K non-zero (unit) elements per row and $C(=KM/N)$ per column while B has L per row/column. The matrix $B^{-1}A$ is then used for encoding the message

$$\boldsymbol{t}_B = B^{-1}A\boldsymbol{s} \pmod 2.$$

The Boolean message vector $\boldsymbol{t}_B$ is then transmitted as a vector $\boldsymbol{t}$ of *real-valued* elements, which we will choose for simplicity as ± 1, and is corrupted by a real-valued noise vector $\boldsymbol{\nu}$, where each element is sampled from a Gaussian distribution of zero mean and variance σ^2. The received message is of the form

$$\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{\nu}.$$

Using the noise model and the probability of the transmitted bit being $t_\mu = \pm 1$ given the received value r_μ:

$$P(t_\mu = \pm 1|r_\mu) = \frac{\mathrm{e}^{-(t_\mu - r_\mu)^2/2\sigma^2}}{\mathrm{e}^{-(t_\mu - r_\mu)^2/2\sigma^2} + \mathrm{e}^{-(t_\mu + r_\mu)^2/2\sigma^2}} = \frac{1}{1 + \mathrm{e}^{-2t_\mu r_\mu/\sigma^2}} \tag{3}$$

one can easily convert the real-valued noise $\boldsymbol{\nu}$ to a flip noise vector such that the probability of an error $n_\mu = 1$ (error) is

$$P(n_\mu = 1) = \frac{1}{1 + \mathrm{e}^{2t_\mu r_\mu/\sigma^2}} \tag{4}$$

where $t_\mu = \pm 1$ is the actual transmitted value. Note that $P(n_\mu = 1)$ may be larger than $\frac{1}{2}$. The noise vector $\boldsymbol{n}$ and our estimate for the transmitted vector $\hat{\boldsymbol{t}}$ are defined probabilistically by using the probabilities derived in (4) and (3), respectively.

Having an estimate for the transmitted vector $\hat{\boldsymbol{t}}$ as well as an estimate for the noise vector $\boldsymbol{n}$, one decodes the binary received message $\hat{\boldsymbol{t}}$ by employing the matrix B to obtain

$$\boldsymbol{z} = B\hat{\boldsymbol{t}} = A\boldsymbol{s} + B\boldsymbol{n}. \tag{5}$$

This requires solving the equation

$$[A, B]\begin{bmatrix} \boldsymbol{s}' \\ \boldsymbol{n}' \end{bmatrix} = \boldsymbol{z}$$

where $\boldsymbol{s}'$ and $\boldsymbol{n}'$ are the unknowns. Solving the equations is being carried out here using methods of belief network decoding [4, 14], where pseudo-posterior probabilities, for the decoded message bits being 0 or 1, are calculated by solving iteratively a set of equations for the conditional probabilities of the codeword bits given the decoded message and vice versa. For exact details of the method used and the equation themselves see [4]. Two differences from the framework used in the case of a BSC should be noted. (a) The probabilities of (4) and (3) may be used for defining the priors for *single components* of the noise and signal vectors respectively. (b) Initial conditions for the noise part of the dynamics may also be derived using (4).

The key point in obtaining improved performance is the construction of the matrices A and B. The original MN code [4] as well as that of Gallager [3] advocated the use of regular architectures with fixed column connectivity; it was also suggested that fixed K values may be preferred. Recent work in the area of irregular codes [11–13] suggest that irregular codes have the potential of providing superior performance which nearly saturates Shannon's limit. These methods concentrate on different column connectivities and use high K and C values (up to 50), which of course increase the complexity of the algorithm and the decoding time required. Decoding delays are of major consideration in most practical applications.

Our method uses the same structure as the MN codes and builds on insight gained from the study of physical systems with symmetric and asymmetric [16] multi-spin interactions and from examining special cases of Gallager's method [7, 9]. Our previous studies for the binary symmetric channel [5] suggest that a careful construction, based on different K and L values for the sub-matrices of A and B respectively, while keeping the connectivity of each of the sub-matrices (and of the matrix as a whole) as uniform as possible, will provide the best results. The guidelines for this architecture are given below and come from the mean-field calculations of [5, 17], showing that the choice of low K and L value codes results in a large basin of attraction but imperfect end magnetization, while codes with higher K and L values can potentially saturate Shannon's bound but suffer from a rapidly decreasing basin of attraction as K and L increase. To exploit the advantages of both architectures and obtain optimal performance, a cascading method was suggested [5, 17] whereby one constructs the matrices A and B from sub-matrices of different K and L values, such that lower values will drive the overlap increase between the decoded and the original messages to a level that enables the higher connectivity sub-matrices to come into play, allowing the system to converge to the perfectly decoded message [17].

Optimizing the trade-off between having a large basin of attraction and improved end magnetization can be done straightforwardly [17] in the case of simple codes [6], but is not very easy in general. Guidelines for optimizing the construction in the general case have been provided in [5]; the key points include: (1) the first sub-matrices are characterized by low K and L values ($\leqslant 2$), while K values in subsequent sub-matrices are chosen gradually higher,

so as to support the correction of faulty bits, and $L = 1$; (2) keeping the number of non-zero column elements as uniform as possible (preferably fixed); (3) to guarantee the inversion of the matrix B, and since noise bits have no explicit correlation, we use a patterned structure, $B_{i,k} = \delta_{i,k} + \delta_{i,k+5}$, for the B sub-matrices with $L = 2$ and $B_{i,k} = \delta_{i,k}$ for $L = 1$; (4) the sub-matrix with the lowest K value, which dominated the dynamics in the initial low-magnetization stage has to include some odd K values in order to break the inversion symmetry, otherwise the two solutions with $m = \pm 1$ are equally attractive. It was also found to dramatically improve the convergence times.

We will now focus on two specific architectures, constructed for the cases of $R = \frac{1}{2} = \frac{1}{4}$, to demonstrate the exceptional performance obtained by employing this method. In each of the cases we divided the composed matrix $[A|B]$ into several sub-matrices characterized by specific K and L values as explained in table 1; the dimensionalities of the full A and B matrices are $M \times N$ and $M \times M$, respectively. Sub-matrix elements were chosen at random (in matrix A) according to the guidelines mentioned above. Encoding was carried out straightforwardly by using the matrix $B^{-1}A$. The corrupted messages were decoded using the set of recursive equations of [4], using random initial conditions for the signal while the initial conditions for the noise vector where obtained according to the noise and signal probabilities equation (4). The prior probabilities of were chosen according to (4) and (3).

In each experiment, T blocks of N-bit unbiased messages were sent through a Gaussian noisy channel of zero mean and variance σ^2 (enforced exactly); the bit error rate, denoted p_b, was monitored. We performed between $T = 10^4$–5×10^4 trial runs for each system size and noise level, starting from different initial conditions. These were averaged to obtain the mean bit error rate and the corresponding variance. In most of our experiments we observed convergence after less than 100 iterations, except very close to the critical noise level. The main halting criterion we adopted relies on either obtaining a solution to (5) or by the stationarity of the first N bits (i.e., the decoded message) over a certain number of iterations. One should also mention that the decoding algorithm's complexity is of $O(N)$ as all matrices are sparse. The inversion of the matrix B is carried out only once and requires $O(1)$ operations due to the structure chosen.

The construction used for the matrices in these two cases appear in table 1 as well as the maximal standard deviation σ_c^N for which $P_b < 2 \times 10^{-5}$ for a given message length N, the predicted maximal standard deviation σ_c^∞ once finite size effects have been considered (discussed below) and Shannon's maximal standard deviation σ_c defined in (2). These results, as well as other results reported here, may be improved upon by avoiding matrices with small loops and by replacing the method of belief propagation by belief revision (our random construction of the matrix A even allows for small loops of size one). It was shown that both improvements have a significant impact on the performance of these types of codes [4, 15]. With these improvements, the actual bit error is expected to be typically lower than the reported value of $P_b = 2 \times 10^{-5}$; however, as we have been limited to about $T = 5 \times 10^4$ trials per noise value we can only provide an upper bound to the actual error values.

To compare our results to those obtained by using turbo codes [18] and in [13] we plotted in figure 1 the two curves (dotted and dashed, respectively), for $N = 10^3$ and 10^4, against the results obtained using our cascading connection method (filled triangles). It is clear from the figure that results obtained using our method are superior in all cases examined. Furthermore, from table 1, one can conclude that the averaged connectivities, $\overline{C}$, for values of $R = \frac{1}{2}$ and $\frac{1}{4}$ are 5 and 9 respectively for the matrix A, and $\frac{3}{2}$ for the matrix B. Similarly, the averaged K values for $R = \frac{1}{2}$ and $\frac{1}{4}$ are $\overline{K} = \frac{5}{2}$ and $\frac{9}{4}$, respectively. These figures are much smaller than those used in [12,13] and other irregular constructions. Minimizing $\overline{K}$ and $\overline{C}$ is of great interest

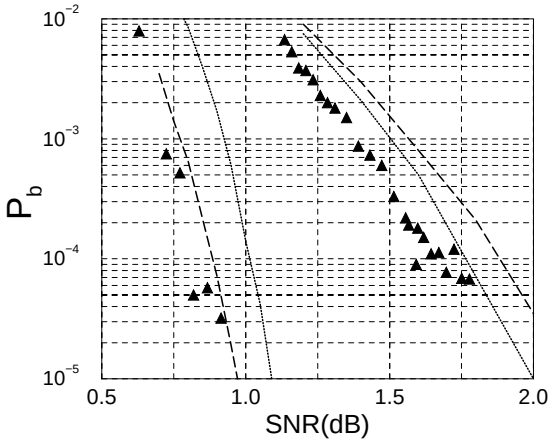


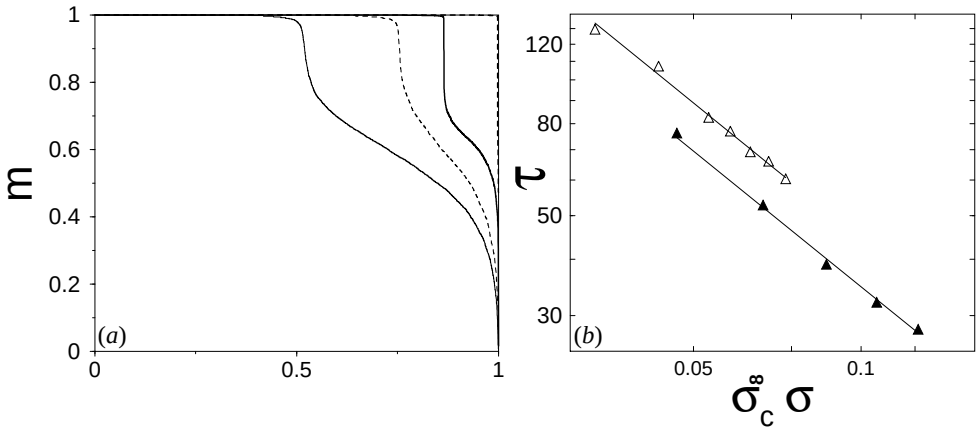
Figure 1. Bit error rate p_b as a function of the standard deviation for a given code rate $R = \frac{1}{2}$ for systems of size $N = 1000, 10\,000$ (right and left, respectively). Our results for each system size appear as black triangles, while results obtained via the turbo code and in [13] for systems of similar sizes appear as curves (dotted and dashed, respectively).

to practitioners since decoding delays are directly proportional to the $\overline{K}$ and $\overline{C}$ values used [4].

It is clear from figure 1 that the finite-size effects are significant in defining the code's performance. It is therefore desirable to find the performance in the limit of infinite messages which are also assumed in deriving Shannon's bound. We employ two main methods for studying the finite-size effects. (a) The transition from perfect ($m(\sigma) = 1$) to no retrieval ($m(\sigma) = 0$), as a function of the standard deviation σ, is expected to become a step function (at σ_c^∞) as $N \to \infty$; therefore, if the percentage of perfectly retrieved blocks in the sample, for a given standard deviation σ, increases (decreases) with N one can deduce that $\sigma < \sigma_c^\infty$ (or $\sigma > \sigma_c^\infty$). (b) Convergence times near criticality usually diverge as $1/(\sigma_c^\infty - \sigma)$; by monitoring average convergence times for various σ values and extrapolating one may deduce the corresponding critical standard deviation.

Both methods have been used in finding the critical values for $R = \frac{1}{2}$ and $\frac{1}{4}$; the results obtained appear in table 1. In figure 2 we demonstrate the two methods: in (*a*) we ordered the samples obtained for $R = \frac{1}{2}$, $\sigma = 0.915, 0.935$ (dashed and solid curves respectively) and $N = 1000, 10\,000$ (thin and thick curves respectively) according to their magnetization; results with higher magnetization appear on the left and the x-axis was normalized to represent fractions of the complete set of trials. One can easily see that the fraction of perfectly retrieved blocks increases with system size indicating that $\sigma < \sigma_c^\infty$. In figure 2(*b*) one finds log–log plots of the mean convergence times τ for $R = \frac{1}{2}, \frac{1}{4}$ and $N = 10\,000$ carried out on perfectly retrieved blocks with less than three error bits. The optimal fitting of expressions of the form $\tau \propto 1/(\sigma_c^\infty - \sigma)$ provides another indication for the σ_c^∞ values, which are consistent with those obtained by the first method.

We end this paper by discussing the main difference between our method and those presented in [11–13]. Firstly, our construction builds on sub-matrices of different K and L values keeping the connectivity in each of the columns as uniform as possible; this equates the corrections received by the various bits while allowing them to participate in different multi-spin interactions, so as to provide contributions of different types throughout the dynamics. In contrast, other irregular codes build on the use of different column connectivities such that a small number of bits, of high connectivity, will lead the decoding process, gathering more



(*a*) (*b*)

Figure 2. (*a*) The block magnetization profile for $R = \frac{1}{2}$, $\sigma = 0.915, 0.935$ (dashed and solid curves respectively) and $N = 1000, 10\,000$ (thin and thick curves respectively), showing the sample magnetization m versus the fraction of the complete set of trials. A total of about 10 000 trials were rearranged in a descending order according to their magnetization values. One can see that the fraction of perfectly retrieved blocks increases with system size. (*b*) Log–log plots of mean convergence times τ for $N = 10\,000$ and $R = \frac{1}{2}, \frac{1}{4}$ (white and black triangles, respectively). The σ_c^∞ values were calculated by fitting expressions of the form $\tau \propto 1/(\sigma_c^\infty - \sigma)$ through the data.

Table 1. The critical noise standard deviation σ_c^N and σ_c^∞ obtained by employing our method for various code rates in comparison with the maximal standard deviation σ_c provided by Shannon's bound. Details of the specific architectures used and their row/column connectivities are also provided.

R	A	K	B	L	$\sigma_c^{10\,000}$ (dB)	σ_c^∞ (dB)	σ_c (dB)
$\frac{1}{2}$	$\frac{1}{10}N \times N$	1	$\frac{1}{10}N \times 2N$	2	0.89	0.973	0.979
	$\frac{9}{10}N \times N$	2	$\frac{9}{10}N \times 2N$	2	(1.012)	(0.238)	(0.185)
	$\frac{3}{4}N \times N$	2	$\frac{3}{4}N \times 2N$	1			
	$\frac{3}{20}N \times N$	6	$\frac{3}{20}N \times 2N$	1			
	$\frac{1}{10}N \times N$	7	$\frac{1}{10}N \times 2N$	1			
$\frac{1}{4}$	$\frac{3}{2}N \times N$	1	$\frac{3}{2}N \times 4N$	2	1.45	1.537	1.550
	$\frac{1}{2}N \times N$	4	$\frac{1}{2}N \times 4N$	2	(-0.217)	(-0.721)	(-0.797)
	$\frac{1}{3}N \times N$	4	$\frac{1}{3}N \times 4N$	1			
	$\frac{5}{6}N \times N$	3	$\frac{5}{6}N \times 4N$	1			
	$\frac{5}{6}N \times N$	2	$\frac{5}{6}N \times 4N$	1			

corrected bits as the decoding progresses. Secondly, [11–13] as well as others point to the need for high multi-spin interactions for achieving performance close to Shannon's bound; we show here that low K, L and C values are sufficient for near-optimal performance (for values of $R = \frac{1}{2}$ and $\frac{1}{4}$ the averaged connectivities are $\overline{C} = 5$ and 9 respectively, for the matrix A, and $\frac{3}{2}$ for the matrix B), allowing one to carry out the encoding and decoding tasks significantly faster. Our work suggests that it is possible to come very close to saturating Shannon's bound with finite connectivity, at least for the code rates considered here. It is plausible that operating close to $R = 1$ will require higher K, L values and may require infinite C or $\overline{C}$ values; this question is currently under investigation.

We have shown that through a successive change in the number of multi-spin interactions (K and L) one can boost the performance of Gallager-type error-correcting codes. The results obtained here for the case of additive Gaussian noise suggests competitive performance to similar state-of-the-art codes for finite-N values; extending the results to the case of infinitely large systems suggest that the current code is less than 0.1 dB from saturating the theoretical bounds set by Shannon. It would be interesting to examine methods for improving the finite-size behaviour of these type of codes; these would be of great interest to practitioners.

Acknowledgment

We would like to thank Dr Yoshiyuki Kabashima for helpful discussions.

References

[1] Shannon C E 1948 *Bell Syst. Tech. J.* **27** 379
Shannon C E 1948 *Bell Syst. Tech. J.* **27** 623
[2] Michelson A M and Levesque A H 1985 *Error-Control Techniques for Digital Communications* (New York: Wiley)
[3] Gallager R G 1963 *Low Density Parity Check Codes (Research Monograph Series 21)* (Cambridge, MA: MIT Press)
Gallager R G 1962 *IRE Trans. Inf. Theory* **IT-8** 21
[4] MacKay D J C and Neal R M 1997 *Electron. Lett.* **33** 457
MacKay D J C 1999 *IEEE Trans. Inf. Theory* **45** 399
[5] Kanter I and Saad D 1999 *Phys. Rev. Lett.* **83** 2660
[6] Sourlas N 1989 *Nature* **339** 693
[7] Kabashima Y and Saad D 1999 *Eur. Phys. Lett.* **45** 97
[8] Saakian D B 1998 *JETP Lett.* **67** 440
[9] Kabashima Y, Murayama T and Saad D 2000 *Phys. Rev. Lett.* **84** 1355
[10] Cover T M and Thomas J A 1991 *Elements of Information Theory* (New York: Wiley)
[11] Mackay D J C, Wilson S T and Davey M C 1999 *IEEE Trans. Commun.* **47** 1449
[12] Luby M, Mitzenmacher M, Shokrollahi A and Spielman D 1998 *IEEE Proc. Int. Symp. on Information Theory ISIT98*
[13] Richardson T, Shokrollahi A and Urbanke R 1999 unpublished
[14] Frey B J 1998 *Graphical Models for Machine Learning and Digital Communication* (Cambridge, MA: MIT Press)
[15] Weiss Y 2000 *Neural Comput.* **12** 1
[16] Kanter I 1998 *Phys. Rev.* A **38** 5972
Kanter I and Sompolinsky H 1987 *Phys. Rev. Lett.* **58** 164
[17] Kanter I and Saad D 2000 *Phys. Rev.* E **61** 2137
[18] Berrou C, Glavieux A and Thitimajshima P 1993 *Proc. 1993 IEEE Int. Conf. on Communications (Geneva, Switzerland, 1993)* vol 1064
Disvilar D and Pollara F 1995 *JPL Pasadena Technical Report* TDA 42