ITW 1998, Killarney, Ireland, June 22 – 26

Low Density Parity Check Codes over $GF(q)$

Matthew C. Davey and David J. C. MacKay
Cavendish Laboratory, Cambridge CB3 OHE, England,
mcdavey@mrao.cam.ac.uk, mackay@mrao.cam.ac.uk.

Abstract — **Binary Low Density Parity Check (LDPC) codes have been shown to have near Shannon limit performance when decoded using a probabilistic decoding algorithm. The analogous codes defined over finite fields $GF(q)$ of order $q > 2$ show significantly improved performance. We present the results of Monte Carlo simulations of the decoding of infinite LDPC Codes which can be used to obtain good constructions for finite Codes. We also present empirical results for the Gaussian channel including a rate 1/4 code with bit error probability of 10^{-4} at $E_b/N_0 = -0.05$dB.**

I. Introduction

We consider a class of error correcting codes first described by Gallager in 1962 [2]. These recently rediscovered low density parity check (LDPC) codes are defined in terms of a sparse parity check matrix and are known to be asymptotically good for all channels with symmetric stationary ergodic noise [6]. Practical decoding of these codes is possible using an approximate belief propagation algorithm and near Shannon limit performance has been reported [7].

We consider the generalisation of binary LDPC codes to finite fields $GF(q)$, $q > 2$, and demonstrate a significant improvement in empirical performance. Although little is known about the theoretical properties of the approximate belief propagation algorithm when applied to this decoding problem, Monte Carlo methods may be used to simulate the behaviour of the decoding algorithm applied to an infinite LDPC code. We have used such Monte Carlo results to design better codes for practical decoding.

In section II we define low density parity check codes and in section III we describe the decoding algorithm. Section IV presents the results of the Monte Carlo simulation and empirical decoding results are presented in section V.

II. Code Construction

The codes are defined in terms of a low density parity check matrix $\mathbf{H}$ as follows. We choose a source block length K, a transmitted block length N and a mean column weight $t > 2$. The *weight* of a vector is the number of non-zero components in that vector. We define $M = (N - K)$ to be the number of parity checks in the code. $\mathbf{H}$ is a rectangular matrix with M rows and N columns. We construct $\mathbf{H}$ such that the weight of each column is at least 2, the mean column weight is t and the weight per row is as uniform as possible.

We fill the non-zero entries in $\mathbf{H}$ from the elements of a finite field $GF(q)$, $q = 2^b$, according to a carefully selected random distribution: rather than using the uniform distribution we choose the entries in each row to maximise the entropy of the corresponding bit of the syndrome vector $\mathbf{z} = \mathbf{Hx}$ where $\mathbf{x}$ is a sample from the assumed channel noise model. Although the code construction is largely random, we may reduce the probability of introducing low weight codewords by constructing the weight 2 columns systematically. To generate codewords we would derive the generator matrix using Gaussian elimination.

If the rows of $\mathbf{H}$ are not independent (for odd t, this has small probability) $\mathbf{H}$ is a parity check matrix for a code with the same N and with smaller M. So $\mathbf{H}$ defines a code with rate of *at least* K/N.

III. Decoding algorithm

We transmit a vector $\mathbf{z}$ which is received as $\mathbf{r} = \mathbf{z} + \mathbf{n}$ where $\mathbf{n}$ is a sample from the channel noise distribution. An instance of the decoding problem requires finding the most probable vector $\mathbf{x}$ such that $\mathbf{Hx} = \mathbf{z}$, where $\mathbf{z}$ is the syndrome vector $\mathbf{z} := \mathbf{Hr} = \mathbf{Hn}$ and the likelihood of $\mathbf{x}$ is determined by the channel model. The decoding algorithm we use is a generalisation of the approximate belief propagation algorithm [8] used by Gallager [2] and MacKay and Neal [7, 6]. The complexity of decoding scales as Ntq^2 per iteration.

We refer to elements of $\mathbf{n}$ as *noise symbols* and elements of $\mathbf{z}$ as *checks*. The belief propagation algorithm may be viewed as a message passing algorithm on a directed bipartite graph defined by the parity check matrix $\mathbf{H}$. Each node is associated with a check or a noise symbol. Let edge e_{ij} connect check i with noise symbol j. For each edge e_{ij} in the graph quantities q^a_{ij} and r^a_{ij} are iteratively updated. q^a_{ij} approximates the probability that the jth element of $\mathbf{x}$ is a, given the information obtained from all checks other than i. r^a_{ij} approximates the probability that the ith check is satisfied if element j of $\mathbf{x}$ is a and the other noise symbols have a separable distribution given by the appropriate $q_{ij'}$. We initialise the algorithm by setting the q^a_{ij} to the likelihood that the jth element of $\mathbf{x}$ is a, as given by the channel model.

After each iteration we make a tentative decoding $\hat{\mathbf{x}}$ by choosing, for each element, the noise symbol that receives the largest vote from the r^a_{ij} messages. If $\mathbf{H}\hat{\mathbf{x}} = \mathbf{z}$ then the decoding algorithm halts having identified a valid decoding of the syndrome, otherwise the algorithm repeats. A failure is declared if some maximum number of iterations (*e.g.* 500) occurs without a valid decoding.

In the absence of loops the algorithm would converge to the correct posterior distribution over noise vectors $\mathbf{x}$. In the presence of cycles convergence is not guaranteed, but decoding performance proves to be very good.

IV. Monte Carlo simulation

We can use Monte Carlo methods to simulate an infinite LDPC code in order to investigate the properties of the decoding algorithm in the absence of loops. We would expect the behaviour to be similar to the empirical performance as the blocklength of the codes increased. We form an ensemble of noise symbols $\mathcal{N} := \{n_i, \{Q^a_i\}_{a \in GF(q)}\}\}^N_1$ emitting initial messages Q_i^a according to our channel model. We then form a fragment of an infinite graph using this ensemble, reflecting our matrix construction, and propagate the Q messages down to a new noise symbol to produce an element of a new ensemble $\mathcal{N}'$. This ensemble $\mathcal{N}'$ will represent an ensemble of noise symbols after one iteration of the decoding algorithm. We iterate the procedure to produce successive ensembles containing approximations to the distribution of Q messages in an infinite network after an arbitrary number of iterations. For successful decoding the average Shannon entropy of the Q messages should become arbitrarily small as decoding progresses.

We declare a decoding 'success' if the average entropy of the Q messages in our ensemble drops below some chosen threshold. With this approach it is possible to investigate the effect of changes in field order, code construction and noise level on the decoding performance.

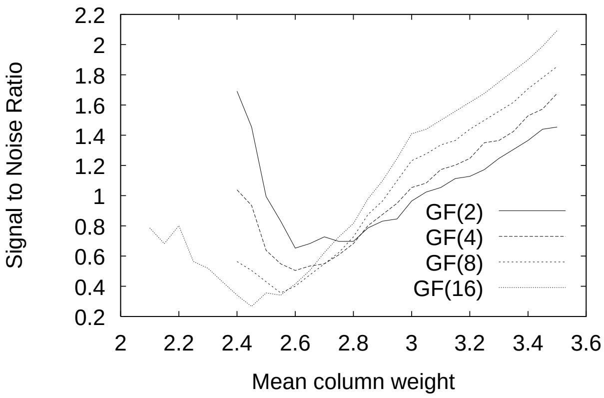

Fig. 1: Gaussian Channel: Minimum signal to noise ratio for which average bit entropy reaches 10^{-5} within 80 iterations, as a function of mean column weight. Rate 1/4 code.

In Figure 1 we present results for the Binary Gaussian Channel. The results suggest that for best performance we should choose the highest order code that is feasible, bearing in mind that the decoding time per iteration scales as q^2. We then choose an appropriate mean column weight, which will be lower as the order increases.

V. Empirical Results

We have used the results of the MC analysis to construct finite codes with performance very close to the Shannon limit [1]. We have produced rate 1/4 codes with mean column weight 2.3 with bit error probability of 10^{-5} at $E_b/N_0 = 0.2$dB.

Recent work by Spielman *et.al.*[4, 5] showed that carefully constructed irregular parity check matrices (non-uniform weight per column) could give improved performance for codes over binary fields. Preliminary results for irregular matrices defined over $GF(8)$ and $GF(4)$ have produced very encouraging results, presented in figure 2. We include a code with bit error rate of 10^{-4} at $E_b/N_0 = -0.05$dB, a slight improvement over the best turbo codes of which this author is aware.

We find that the codes presented in [4], with mean column weight 8, perform poorly over higher order fields. Motivated by the Monte Carlo results, we tried irregular constructions with quite low weight. The best code in figure 2 has a mean column weight of 3.4, but contains column weights of up to 33, as shown in table 1. The row weight is almost uniform.

Col. Weight	2	3	9	13	17	33
fraction	0.67	0.23	0.04	0.03	0.02	0.01

Tab. 1: Parameters of good irregular code for $GF(16)$, rate 0.25

No investigation of the parameters of irregular LDPC codes has yet been performed using MC methods, but we expect more careful choice of code parameters to yield further improvements.

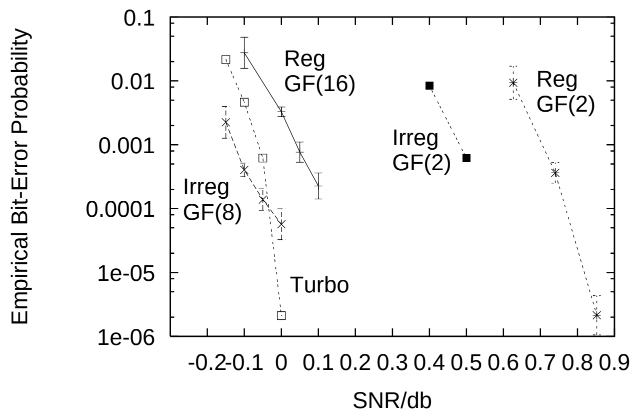

Fig. 2: Empirical results for Gaussian Channel, Rate 1/4 Left-Right : Irregular LDPC, $GF(8)$ blocklength 24000 bits; JPL Turbo [3]; Regular LDPC, $GF(16)$, blocklength 24448 bits; Irregular LDPC , $GF(2)$, blocklength 64000 bits[4]; Regular LDPC, $GF(2)$, blocklength 40000 bits[6]

References

[1] M. C. Davey and D. J. C. MacKay. Low density parity check codes over GF(q). *IEEE Communications Letters*, June 1998.

[2] R. G. Gallager. Low density parity check codes. *IRE Trans. Info. Theory*, IT-8:21–28, Jan 1962.

[3] JPL. Turbo codes performance. Available from `http://www331.jpl.nasa.gov/public/TurboPerf.html`, August 1996.

[4] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Improved low–density parity–check codes using irregular graphs and belief propagation. Submitted to ISIT98, 1998.

[5] M. G. Luby, M. Mitzenmacher, M. Amin Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing (STOC)*.

[6] D. J. C. MacKay. Good error correcting codes based on very sparse matrices. Submitted to IEEE transactions on Information Theory. Available from `http://wol.ra.phy.cam.ac.uk/`, 1997.

[7] D. J. C. MacKay and R. M. Neal. Near Shannon limit performance of low density parity check codes. *Electronics Letters*, 32(18):1645–1646, August 1996. Reprinted *Electronics Letters*, vol 33, no 6, 13th March 1997, p.457–458.

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