Design of Provably Good Low-Density Parity Check Codes

Tom Richardson, Amin Shokrollahi and Rüdiger Urbanke
Bell Labs, Lucent Technologies
Murray Hill, NJ 07974

April 5, 1999

Abstract

We design sequences of low-density parity check codes that provably perform at rates extremely close to the Shannon capacity. The codes are built from highly irregular bipartite graphs with carefully chosen degree patterns on both sides. Our theoretical analysis of the codes is based on [1]. Additionally, based on the assumption that the underlying communication channel is symmetric, we prove that the probability densities at the message nodes of the graph satisfy a certain symmetry. This enables us to derive a succinct description of the density evolution for the case of a belief propagation decoder. Furthermore, we prove a stability condition which implies an upper bound on the fraction of errors that a belief propagation decoder can correct when applied to a code induced from a bipartite graph with a given degree distribution.

Our codes are found by optimizing the degree structure of the underlying graphs. We develop several strategies to perform this optimization. We also present some simulation results for the codes found which show that the performance of the codes is very close to the asymptotic theoretical bounds.

Index Terms Low density parity check codes, belief propagation, turbo codes, irregular codes

1 Introduction

In this paper we present *irregular* low-density parity check codes (LDPCCs) which exhibit performance extremely close to the best possible as determined by the Shannon capacity formula. For the additive white Gaussian noise channel (AWGNC) the best code of rate one-half presented in this paper has a threshold within 0.06dB from capacity, and simulation results show that our best LDPCC of length 10^6 achieves a bit error probability of 10^{-6} less than 0.13dB away from capacity, handily beating the best (turbo) codes known so far.

LDPCCs possess several other distinct advantages over turbo-codes. First, the complexity of (belief-propagation) decoding is somewhat less than that of turbo-codes and, being fully parallelizable, can potentially be performed at significantly greater speeds. Second, as indicated in a previous paper [1], very low complexity decoders that closely approximate belief-propagation in performance may

be (and have been) designed for these codes. Third, low-density parity check decoding is verifiable in the sense that decoding to a correct codeword is a detectable event. One practical objection to low-density parity-check codes has been that their encoding complexity is high. One way to get around this problem is by slightly modifying the construction of codes from bipartite graphs to a cascade of such graphs, see [2, 3]. A different solution for practical purposes, which does not require cascades, will be presented elsewhere [4].

Let us recall some basic notation. As is well known, low-density parity-check codes are well represented by bipartite graphs in which one set of nodes, the *variable nodes*, corresponds to elements of the codeword and the other set of nodes, the *check nodes*, corresponds to the set of parity-check constraints satisfied by codewords of the code. *Regular* low-density parity-check codes are those for which all nodes of the same type have the same degree. Thus, a (3,6)-regular low-density parity-check code has a graphical representation in which all variable nodes have degree 3 and all check nodes have degree 6. The bipartite graph determining such a code is shown in Fig. 1.

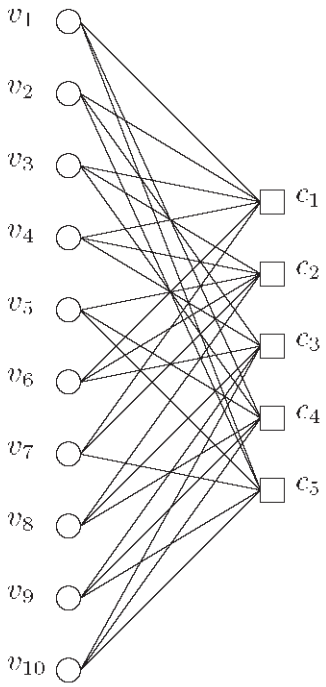

Figure 1: A $(3, 6)$-regular code of length 10. There are 10 variable nodes and 5 check nodes. For each check node c_i the sum (over GF(2)) of all adjacent variable nodes is equal to zero.

For an *irregular* low-density parity-check code the degrees of each set of nodes are chosen according to some distribution. Thus, an irregular low-density parity-check code might have a graphical

representation in which half the variable nodes have degree 5 and half have degree 3, while half the constraint nodes have degree 6 and half have degree 8. For a given length and a given degree sequence (finite distribution) we define an *ensemble* of codes by choosing the edges, i.e., the connections between variable and check nodes, randomly. More precisely, we enumerate the edges emanating from the variable nodes in some arbitrary order and proceed in the same way with the edges emanating from the check nodes. Assume that the number of edges is E. Then a code (a particular instance of this ensemble) can be identified with a permutation on E letters. Note that all elements in this ensemble are equiprobable. In practice the edges are not chosen entirely randomly since certain potentially unfortunate events in the graph construction can be easily avoided.

In a recent paper [1] we presented an asymptotic analysis of LDPCCs under message passing decoding. Assume we have the following setup:

1. We are given an ordered family of binary-input discrete memoryless channels parameterized by a real parameter δ such that if $\delta_1 < \delta_2$ then the channel with parameter δ_2 is a physically degraded version of the channel with parameter δ_1. Further, each channel in this family fulfills the channel symmetry condition

$$p(y|x = 1) = p(-y|x = -1) \; . \tag{1}$$

2. A sequence of code ensembles $\mathcal{C}_n(\lambda, \rho)$ is specified, where n is the length of the code and $\lambda(x) := \sum_{i=1}^{d_l} \lambda_i x^{i-1}$ ($\rho(x) := \sum_{i=1}^{d_r} \rho_i x^{i-1}$) is the variable (check) node degree sequence. More precisely, λ_i (ρ_i) is the fraction of edges emanating from variable (check) nodes of degree i. Note that the rate of the code is given in terms of $\lambda(x)$ and $\rho(x)$ as $1 - \frac{\int_0^1 dx\ \rho(x)}{\int_0^1 dx\ \lambda(x)}$.

3. A message passing decoder is selected. By definition, messages only contain *extrinsic* information, i.e., the message emitted along an edge e does not depend on the incoming message along the same edge. Further, the decoder fulfills the following symmetry conditions. Flipping the sign of all incoming messages at a variable node results in a flip of the sign of all outgoing messages. The symmetry condition at a check node is slightly more involved. Let e be an edge emanating from a check node c. Then flipping the sign of i incoming messages arriving at node c, excluding the message along edge e, results in a change of the sign of the outgoing message by $(-1)^i$. In all these cases, only the sign is changed, but the reliability remains unchanged.

Under the above assumptions there exists a threshold δ^* with the following properties: For any $\epsilon > 0$ and $\delta < \delta^*$ there exists an $n(\epsilon, \delta)$ such that almost every code[1] in $\mathcal{C}_n(\lambda, \rho)$, $n > n(\epsilon, \delta)$, has probability of error smaller than ϵ assuming that transmission takes place over a channel with parameter δ. Conversely, if the transmission takes place over a channel with parameter $\delta > \delta^*$, then almost any code in $\mathcal{C}_n(\lambda, \rho)$ has probability of error uniformly bounded away from zero.[2]

Further, for the important case of belief propagation decoders a procedure was introduced that enables the efficient computation of δ^* to any desired degree of accuracy. In [1] threshold values and simulation results were given for a variety of noise models, but the class of low-density parity-check codes considered was largely restricted to *regular* codes. In the present paper we present results indicating the remarkable performance that can be achieved by properly chosen *irregular* codes.

The idea underlying this paper is quite straightforward. Assume we are interested in transmission over a particular memoryless channel using a particular message passing decoder. Since for every given pair of degree sequences (λ, ρ) we can determine the resulting threshold value δ^*, it is natural to search for those pairs which maximize this threshold.[3] This was done, with great success, in the case of erasure channels [5, 6, 7]. In the case of most other channels of interest the situation is much more complicated and new methods must be brought to bear on the optimization problem. Fig. 2 compares the performance of the $(3, 6)$-regular LDPCC (which is the best regular such code) with the performance of the best irregular LDPCC we found in our search and the performance of the standard parallel turbo code introduced by Berrou, Galvieux, and Thitimajshima [8]. All three codes have rate one-half and their performance under belief propagation decoding over the AWGNC is shown for a code word length 10^6. Also shown is the Shannon limit and the threshold value of our best LDPCC ($\sigma^* = 0.9718$). From this figure it is clear how much benefit can be derived from optimizing the degree sequences. For $n = 10^6$ and a bit error probability of 10^{-6}, our best LDPCC is only 0.13dB away from capacity. This handily beats the performance of turbo-codes. Even more impressive, the threshold, which indicates the performance for infinite lengths, is a mere 0.06dB away from capacity.

[1] More precisely, the fraction of codes for which the above statement is true converges exponentially (in n) fast to 1.

[2] We conjecture that this is actually true for *every* code in $\mathcal{C}_n(\lambda, \rho)$.

[3] We may also optimize degree sequences under various constraints. For example, the larger the degrees used the larger the code size needs to be in order to approach the predicted asymptote. Therefore, it is highly desirable to look for the best degree sequences with some a priori bound on the size of the degrees.

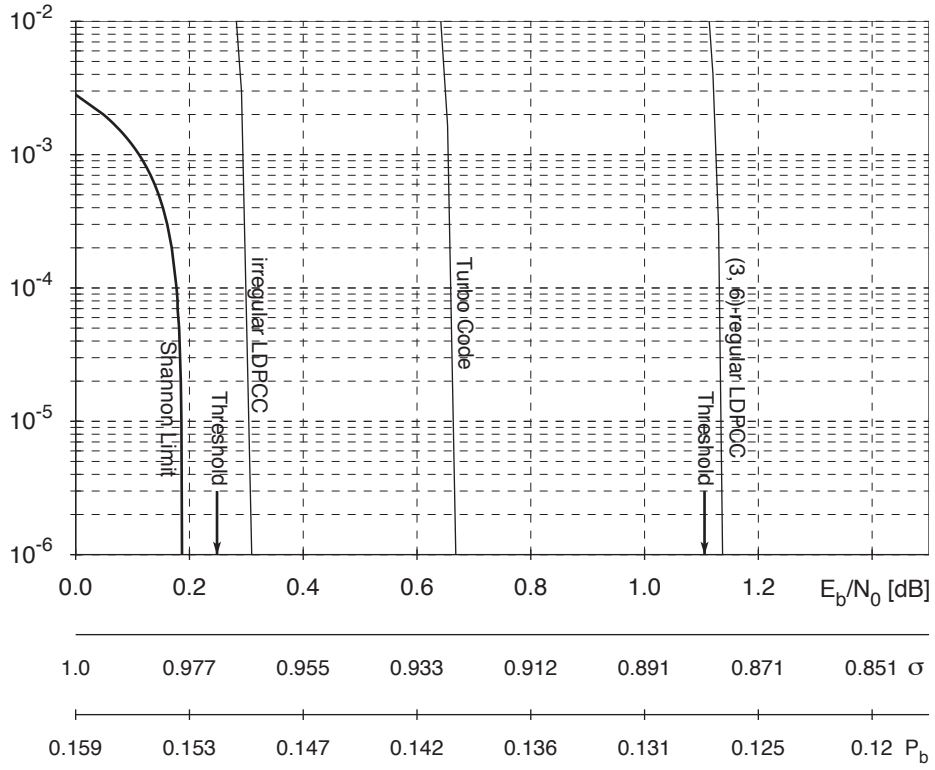

Figure 2: Comparison of $(3,6)$-regular LDPCC, turbo code, and optimized irregular LDPCC. All codes are of length 10^6 and of rate one-half. The bit error rate for the AWGNC is shown as a function of E_b/N_0 (in dB), the standard deviation σ, as well as the raw input bit error probability P_b.

The empirical evidence presented in Fig. 2 together with the results presented in Section 3 beg the question of whether LDPCCs can achieve capacity on a given binary-input memoryless channel. The only result in this direction is that of [5] which gives an explicit sequence of degree distributions such that, in the limit, the codes induced by these degree distributions achieve capacity on an erasure channel. The following Theorem, due to Gallager, imposes, at least for the BSC, a necessary condition in order for LDPCCs to achieve capacity: their maximum right degree d_r must tend to infinity.[4] Although this result bounds the performance of LDPCCs away from capacity, the gap is extremely small and the gap converges to zero exponentially fast in d_r. Hence, although of great theoretical interest, the following theorem does not impose a significant practical limitation.

Theorem 1. [Gallager 61] Let $\mathcal{C} \in \mathcal{C}(\lambda, \rho)$ be a LDPCC of rate r. Let $\mathcal{C}$ be used over a BSC with crossover probability δ and assume that each codeword is used with equal probability. If

[4] We conjecture that a similar statement (and proof) could be given for continuous channels.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.