Enumerators for Protograph Ensembles of LDPC Codes

S. L. Fogal
Caltech
1200 E. California Blvd
Email: sarah@acm.caltech.edu

Robert McEliece
Caltech
1200 E. California Blvd
Email: rjm@systems.caltech.edu

Jeremy Thorpe
Caltech
1200 E. California Blvd
Email: jeremy@systems.caltech.edu

I. Introduction

LDPC codes are becoming a standard in today's error correcting systems. However, even as the number of codes investigated by researchers has swelled, it remains difficult to find codes achieving "near zero" error probability at rates close to Shannon capacity. Instead, codes which are designed to behave well close to the capacity limit typically exhibit an Error floor.

Error floors are generally attributed to small sets of variables such as low-weight codewords, low-weight stopping sets [1], pseudocodewords [2] of small pseudo-weight, or, in the case of quantized decoders, small trapping sets [3]. Often, these sets are discovered only after specific codes have been designed and simulated. However, it is desirable to be able to predict the existence and frequency of such sets for entire ensembles of codes.

Speaking more formally, we are interested in certain asymptotic weight enumerators of LDPC code ensembles. Gallager was able to compute asymptotic codeword weight enumerators for regular LDPC codes at least as early as 1963[4]. Litsyn and Shevelev[5] extended this result to include unstructured irregular ensembles. More recently, Di[6] has computed weight enumerators and stopping set enumerators also for unstructured irregular ensembles (in both average and typical case).

In this paper, we consider the problem of finding average enumerators for the class of protograph ensembles, which are related in a certain way to quasi-cyclic codes. Our methods, which are necessarily different from those used to compute enumerators for irregular ensembles, can be applied to both codeword and stopping set weight enumerators, based on their simple combinatorial characterizations.

In section II, we define the quantity $A\Theta, \mathbb{G}$ which is the number of vectors of fractional weight Θ having a certain relationship to the graph $\mathbb{G}$ (e.g. being a codeword or a stopping set). The expectation of this quantity with respect to an ensemble is $\overline{A_N(\Theta)}$. This quantity typically grows exponentially with N and the enumerator exponent $E(\Theta)$ is:

$$E(\Theta) = \lim_{N\to\infty} \frac{1}{N} \ln \overline{A_N(\Theta)} \tag{1}$$

We show that

$$E(\Theta) = \max_{\langle\theta\rangle=\Theta} E(\theta) \tag{2}$$

for a certain function $E(\theta)$ where θ is a vector fractional weight or partial weight. In section IV, we show how to compute the value of $E(\theta)$. In section V, we show that $E(\theta)$ is in general not convex, and thus is difficult to optimize. Nonetheless, we apply steepest ascent to solve the maximization, and show that this method gives results that are reasonable.

In section VI, we outline some future research directions

II. Weight Enumerators Defined

Protograph ensembles are defined and characterized by a bipartite graph $P = (V, C, E)$, where $V = \{v\}$ is a set of variable nodes, $C = \{c\}$ is a set of check nodes, and $E = \{e\}$ is a set of edges each adjacent to one element $v(e) \in V$ and one element $c(e) \in C$. Formally, a protograph is equivalent to a code's Tanner graph, except that multiple edges are allowed.

A protograph P is equivalent to an $r \times n$ protomatrix H where the columns of H are indexed by V, rows are indexed by C, and $H_{c,v}$ is the number of edges in E adjacent to c and v.

The ensemble corresponding to P generates a graph $\mathbb{G}$ of length $N \cdot n$ whose Tanner graph is a random N-lift of P. A random N-lift of P, which we denote $P^N = (V^N, C^N, E^N)$, is constructed from a set of random permutation matrices $\{\pi_e\}_{e\in E}$ each of length N. We let $V^N = V \times \{1..n\}$, $C^N = C \times \{1..n\}$, and $E^N = E \times \{1..n\}$, where (e, i) is adjacent to (c, i) and $(v, \pi_e(i))$. We refer to v as the *type* of node (v, i), and to i as its *index*, using a similar convention for check nodes (c, i) and edges (e, i).

The codewords $x \in \mathbb{G}$ are the assignments of $(0, 1)$ to each $v^N \in V^N$ such that each $(c, i) \in C^N$ is adjacent an even number of times to variable nodes assigned the value 1. Similarly, the stopping sets s of $\mathbb{G}$ are assignments such that each $(c, i) \in C^N$ is adjacent 0 times or at least 2 times to variable nodes assigned the value 1.

We are now ready to define a set Ω which generalizes the notions of codeword and stopping set. For each check $c \in C$ in the protograph define a set of allowed vectors $\Omega_c \subset (0, 1)^{\{e:c(e)=c\}}$.

For a particular word x and check node $(c,i) \in C^N$, define $\omega_{(c,i)}(x)$ to be the vector of variables connected to (c,i):

$$\omega_{(c,i)}(x) = (x_{v(e)})_{e:c(e)=(c,i)} \tag{3}$$

Then the set Ω is the set of x such that every vector $\omega_{(c,i)}(x)$ is in the corresponding allowed set Ω_c, formally:

$$\Omega = \left\{x : \omega_{(c,i)}(x) \in \Omega_c,\, c \in C,\, i \in \{1..n\}\right\} \tag{4}$$

is a set of words x with a certain combinatorial property.

By choosing an appropriate definition of Ω_c, it is possible make Ω the set of codewords or the set of stopping sets. If, for each $c \in C$, Ω_c is the set of vectors of even weight, then Ω is the set of codewords in $\mathbb{G}$. If Ω_c is the set of vectors of weight not equal to 1, then Ω is the set of stopping sets.

We are interested in the number of words in Ω of fractional weight $\Theta(x)$, defined to be the the number of 1's in x divided by the word length $N \cdot n$. For a given graph $\mathbb{G}$ the number of such words is denoted $A(\Theta, \mathbb{G})$, and the expectation with respect to the ensemble of graphs of length N is denoted $\overline{A_N(\Theta)}$This expectation typically grows exponentially with N, and the exponent $E(\Theta)$ is defined by equation 1.

III. Approach

Our approach to computing $E(\Theta)$ is based on the method of types [9]. For a particular word x (not necessarily in Ω), denote its type (or partial weight) by $\theta(x) = (\theta_v)_{v \in V}$, where θ_v denotes the fraction of times that x assigns 1 to variables (v,i) of type $v \in V$. The following lemma says that the probability that $x \in \Omega$ depends only the type θ.

Lemma 1: if x and y are assignments of $(0,1)$ to each $v \in V^N$ such that $\theta(x) = \theta(y)$ then $P(x \in \Omega) = p(y \in \Omega)$

Proof: $\theta_v(x) = \theta_v(y)$ implies that there exists a vector of permutations $(\pi_v)_{v \in V}$ such that for each v, x_v = $\pi_v(y_v)$, where x_v is the value x assigns to $\{v,i\}$.

The permutations $(\pi_v)_{v \in V}$ define a bijection on elements of the ensemble defined by $f((\pi_e)) = (\pi_e \cdot \pi_{(v(e))})$ such that $y \in \Omega_{(f((\pi_e)))}$ if and only if $x \in \Omega_{(\pi_e)}$. Since all lifts in the ensemble are equiprobable, the conclusion holds. ∎

Thus the expected number of words of type θ, which we denote $\overline{A_N(\theta)}$ is just the number of words of type θ times the probability that any word of that type is in Ω.

$$\overline{A_N(\theta)} = |\{x : \theta(x) = \theta\}| \cdot P(x \in \Omega | \theta(x) = \theta) \tag{5}$$

It is straightforward to see that the number of words of type θ can be approximated as:

$$|\{x : \theta(x) = \theta\}| = e^{N \sum_v H(\theta_v)} \tag{6}$$

Define the indicator function that x satisfies all of the constraints

Definition 1:

$$f_{\Omega_c}(x, \mathbb{G}) = \begin{cases} 1, \text{ if } \omega_{(c,i)}(x) \in \Omega_c \forall i \in \{1..n\} \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

The following lemma says that for any x, the probability of satisfying each type of constraint Ω_c is independent over $c \in C$.

Lemma 2:

$$P(x \in \Omega) = \prod_{c \in C} P(f_{\Omega_c}(x, \mathbb{G}) = 1) \tag{8}$$

Proof: for a particular x, $f_{\Omega_c}(x, \mathbb{G})$ is a function only of the set of permutations $\{\pi_e\}_{e:c(e)=c}$. The permutations $\{\pi_e\}$ are mutually independent, and thus independent with respect to the partitioning $\left\{\{\pi_e\}_{e:c(e)=c}\right\}_{c \in C}$. The result follows since functions of independent variables are independent. ∎

We define the asymptotic exponent of the probability that all $\omega_{(c,i)} \in \Omega_c$ for all checks of type c:

$$\Phi_c = \lim_{N \to \infty} ln(P(f_{\Omega_c}(x, \mathbb{G}) = 1))/N \tag{9}$$

For a particular word x, this probability depends only on the vector of weights α_c associated with the variables adjacent to check c in the protograph:

$$\alpha_c = (\theta_{v(e)})_{e:v(e)} \tag{10}$$

From a computational point of view, it is unfortunate that independence does not factor further. Although $f_{\Omega_c}(x, \mathbb{G})$is independent from type to type, $\omega_{(c,i)}(x)$ are generally dependent among values of i. Nonetheless, we can apply large deviation theory and Sanov's theorem to obtain the following theorem, which shows in principle how to compute the asymptotic probability exponent Φ_c.

Theorem 1:

$$\Phi_c = \max_{p \in \mathbb{P}} H(p) - \sum_{e:c(e)=c} H(\theta_{v(e)}) \tag{11}$$

where $\mathbb{P}$ is the set of distributions over Ω_c satisfying the marginal constrains:

$$\sum_{\omega \in \Omega} \omega p(\omega) = \alpha_c \tag{12}$$

Proof: see the appendix. ∎

Taking the log of equation 5, and substituting equation 6 and 9, we have:

$$E(\theta) = \sum_{v \in V} H(\theta_v) + \sum_{c \in C} \Phi_c(\alpha_c) \tag{13}$$

In the following section, we show how to numerically compute the value of Φ_c and thus how to compute $E(\theta)$.

IV. Numerical Methods for computing $E(\theta)$

In the previous section, we have seen that computing each function $\Phi_c(\theta)$ requires solving a constrained entropy maximization problem.

In this section, we describe the computational mathematics used to calculate $E(\Theta)$, for a given protograph described by an $r \times n$ matrix $\mathbf{H}$.

Let m_c be the degree of c. We have seen that Ω_c is a set of m_c-vectors. We seek $\Phi_c(\alpha_c) = \max H(p) - \sum_e H(\theta_v)$ where $p(\omega)$ is the set of all probability mass functions satisfying equation 12

Applying Euler-Lagrange theory, we will see how this constrained optimization problem can be transformed into a non-linear system of equations. The Lagrangian corresponding to our constrained optimization problem can be written

$$\mathcal{L}(p) = -\sum_{\omega \in \Omega} p(\omega)(\log(p(\omega)) - \mathbf{s} \cdot \omega) \tag{14}$$

The constrained optimum must satisfy $\frac{\partial \mathcal{L}}{\partial p} = 0$, and this condition implies a Boltzmann distribution on ω given by

$$p^*(\mathbf{s}, \omega) = \frac{1}{z(\mathbf{s})} e^{-\mathbf{s}\cdot\omega} \tag{15}$$

The normalizing constant $z(\mathbf{s})$ takes the value that ensures p is a probability distribution, appropriately summing to 1.

$$z(vecs) = \sum_{\omega \in \Omega} e^{-\mathbf{s}\cdot\omega}, \tag{16}$$

The Helmholtz free energy can be written in terms of $Z(\mathbf{s})$:

$$F(\mathbf{s}) = -\log(Z(\mathbf{s})) \tag{17}$$

and has the property that its gradient with respect to $\mathbf{s}$ is equal to the l.h.s of equation 12.

$$\nabla F(\mathbf{s}) = \frac{1}{Z(\mathbf{s})} \sum_{\omega \in \Omega} \omega e^{-\mathbf{s}\cdot\omega} \tag{18}$$

$$= \sum_{\omega \in \Omega} \omega p(\omega). \tag{19}$$

Thus, if we find $\mathbf{s}^*$ which solves

$$\nabla F(\mathbf{s}^*) = \alpha_c \tag{20}$$

then the probability density that leads to the maximum entropy is given by $p^*(\mathbf{s}^*, \omega)$ and $\Phi_c(\alpha_c)$ can be expressed

$$\Phi_c(\alpha_c) = -F(\mathbf{s}^*) + \mathbf{s}^* \cdot \nabla F(\mathbf{s}^*) \tag{21}$$

The domain of α_c, usually denoted K, is the convex hull of all $\omega \in \Omega_c$. Outside of this domain, no distribution p can satisfy 12. As a side remark, we note that if each Ω_c is the set of even weighted vectors, then the feasible region is formally equivalent to the pseudocodeword fundamental polytope.

The domain of $\mathbf{s}$ is the entire space R^n of real vectors. It is shown in a concurrently submitted paper by Aji et. al[7] that there is a one-to-one correspondence between these two domains. That result follows from considering the Helmholtz free energy and its Legendre conjugate.

Once we have this non-linear system of equations 20, we can numerically solve for $\mathbf{s}^*$ using either Broyden's or Newton's Method. We found Broyden's Method[8] to be far faster, and thus use it whenever possible. However, since Broyden's Method uses only an approximation to the Jacobian, it is not always able to find a solution. In practice, this typically happens near the boundary of the feasible set(where calculations of gradients and Jacobians become more difficult by any method). The values of α_c on the boundary of K correspond to values of $\mathbf{s}$ with infinite norm, and care must be taken to avoid numerical problems in this region.

In the following section, we will show how to optimize the function $E(\theta)$ over θ to obtain $E(\Theta)$.

V. Optimization of $E(\theta)$

Since there are only polynomially many types, each having an exponential number of elements, it is a standard result that the sum is dominated by a a single type, as expressed in equation 2.

In general, the function $E(\theta)$ is not convex. For protographs with no check nodes of high degree, it may be possible to essentially search the whole space $\{\theta : \langle\theta\rangle = \Theta$ for the global maximum of 2, but this is impractical for protographs with any large check nodes. A second approach is to use a gradient following method such as steepest descent. Unfortunately, this is not guaranteed to converge to the global minimum. A third approach is to use steepest descent starting from a number of different starting locations. In practice, this approach is sufficient to compute curves that appear continuous for protographs that have been investigated.

Still, it is an artifact of certain protographs that there are critical values of Θ at which the global minimum of $E(\theta)$ jumps from one place to another, which is reflected in a discontinuity in the first derivative of $E(\Theta)$.

Figure V shows our evaluation of the weight enumerator for a rate $1/3$ protograph defined by the matrix in equation 22. The zoomed section, shown in figure V shows a discontinuity at approximately $\Theta = 0.13$ in which the global maximum of 2 jumps from one value of θ to another. The dashed lines indicate other local maxima.

$$H = \begin{pmatrix} 3 & 0 & 3 \\ 0 & 3 & 4 \end{pmatrix} \tag{22}$$

For a given matrix, $\mathbf{H}$, and a value of Θ perhaps only some vectors θ have a solution to the entropy maximization problems for each row of $\mathbf{H}$, as for some problems there are no probability mass functions which satisfy the constraints. A simple algorithm can be used to determine whether a vector θ is feasible. In the steepest ascent code, if we find ourself stepping outside the feasible set, we just take a smaller step size until either we remain in the feasible set or the step size becomes effectively zero.

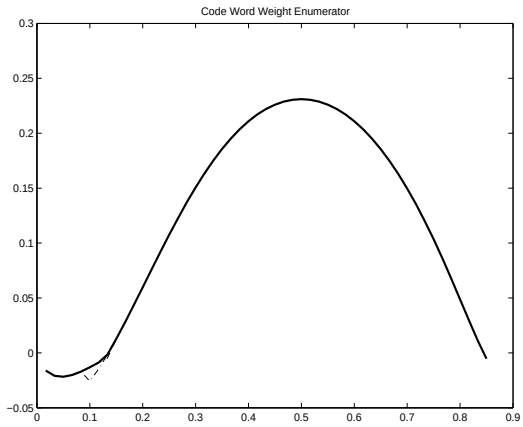

Fig. 1. Numerical evaluation of the weight enumerator for a rate 1/3 protograph

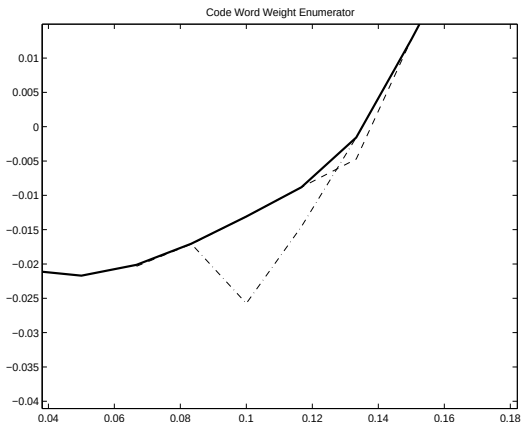

Fig. 2. This weight enumerator has an elbow

VI. Discussion

A primary motivation for computing enumerators has been to use them to design ensembles of codes with low error floors. The general idea is to use a combination of enumerator properties, such as the asymptotic expected minimum weight, and density evolution threshold. Preliminary experiments in which such codes have been designed and simulated have suggested that this approach can be effective.

Multi-Edge-type ensembles are a superclass of both protograph ensembles and unstructured irregular ensembles. Since the theory for computing enumerators for unstructured ensembles is by now established, we hope to be able to combine our techniques with those techniques in order to find enumerators for this class of ensembles.

Appendix

From the definition of the protograph ensemble, the check (c, i) is adjacent to $\{(v(e), \pi_e(i))\}_{e \in E: c(e)=c}$

With m fixed and n very large, denote by $\Omega_n - \Omega_n(\theta_1, \ldots, \theta_m)$ the set of all $m \times n$ $(0,1)$-matrices with row sums $w_1, \ldots, w_m$. There are

$$\prod_{i=1}^{m} \binom{n}{w_i} = \prod_{i=1}^{m} \binom{n}{\theta_i n} \tag{23}$$

(here $\theta_i = w_i/n$, for $i = 1, \ldots, m$.) such matrices, and so we define a uniform probability measure on Ω_n:

$$Q^n(X) = \prod_{i=1}^{m} \binom{n}{w_i}^{-1} \text{ for all } X \in \Omega_n. \tag{24}$$

Let X_j be the j^{th} column of $X \in \Omega_n$. Then $X_1, \ldots, X_n$ are identically distributed (but not independent) V_m-valued random vectors, which we summarize with the notation

$$(X_1, \ldots, X_n) \sim Q^n(X). \tag{25}$$

We denote the common density of the X_j's by $Q^*(\alpha)$:

$$Q^*(\alpha) = \Pr X_j = \alpha = \prod_{i=1}^{m} \theta_i^{\alpha_i} (1-\theta_i)^{1-\alpha_i} \tag{26}$$

for $\alpha = (\alpha_1, \ldots, \alpha_m) \in V_m$

We define the *type* of a matrix $X \in \Omega_n$ as the empirical density on V_m defined by $(X_1, \ldots, X_n)$:

$$P_X(\alpha) = \frac{1}{n} j : X_j = \alpha \text{for } \alpha \in V_m. \tag{27}$$

The set of all such empirical densities will be denoted by ¶$_n$. By C. & T. Theorem 12.1.1,

$$|¶_n| \leq (n+1)^{2^m}. \tag{28}$$

The *type class* of $P \in ¶_n$ is defined as

$$T(P) = \{X \in \Omega_n : P_X = P\} \tag{29}$$

If $X \in \Omega_n$, then clearly (by conservation of 1's):

$$\sum_{\alpha \in V_m} P_X(\alpha)\alpha_i = \theta_i \text{for } i = 1, \ldots, m. \tag{30}$$

A density satisfying 30, empirical or not, is called *consistent*. The set of consistent densities on V_m is denoted by ¶.

Theorem 2: (Cf. Cover and Thomas[9] Theorem 12.1.2.) Let $(X_1, \ldots, X_n) \sim Q^n(X)$. Then to first order in the exponent the probability of X depends only on its type:

$$2^{-n(H(P_X)+D(P_X\|Q^*))} \leq Q^n(X) \leq 2^{-n(H(P_X)+D(P_X\|Q^*))}(n+1)^m.$$

Theorem 3: (Cf. Cover and Thomas Theorem 12.1.4). For any consistent type class $T(P)$,

$$2^{-nD(P\|Q^*)}(n+1)^{-2^m} \leq Q^n(T(P)) \leq 2^{-nD(P\|Q^*)}(n+1)^m.$$

On the other hand, if P is not consistent, $Q^n(T(P)) = 0$.

Theorem 4: (Cf. Cover and Thomas Theorem 12.4.1). Let $(X_1, \ldots, X_n) \sim Q^n(X)$, and let $E \subseteq ¶_n$ be a set of consistent probability distributions. Then

$$(n+1)^{-2^m} 2^{-nD(P^*\|Q^*)} \leq Q^n(E) \leq (n+1)^{2^m+m}\, 2^{-nD(P^*\|Q^*)}$$

where $P^* = \arg\min_{P \in E} D(P\|Q^*)$.

Summary 1: If $X = (X_1, \ldots, X_n) \sim Q^n(X)$, $P \in \P_n$, $E \subseteq \P_n$:

$$\begin{aligned} Q^n(X) &= 2^{-n(H(P_X)+D(P_X\|Q^*)} && (31)\\ Q^n(T(P)) &= 2^{-nD(P\|Q^*)} && (32)\\ \sum_{P\in E} Q^n(T(P)) &= 2^{-nD(P^*\|Q^*)} && (33) \end{aligned}$$

where $P^* = \arg\min_{P\in E} D(P\|Q^*)$.

REFERENCES

[1] C. Di, D. Proietti, E. Telatar, T. Richardson, and R. Urbanke, "Finite length analysis of low-density parity-check codes." Submitted IEEE Trans. on Information Theory, 2001.

[2] B. J. Frey, R. Koetter, and A. Vardy, "Skewness and pseudocodewords in iterative decoding,"

[3] T. Richardson, "Error floors of ldpc codes." 2004.

[4] R. Gallager, "Low-density parity-check codes."

[5] S. Litsyn and V. Shevelev, "On ensembles of low-density parity-check codes: Asymptotic distance distributions.," *IEEE Transactions on Information Theory*, vol. 48, no. 4, pp. 887–908, 2002.

[6] C. Di, "Asymptotic and finite-length analysis of low-density parity-check codes," 2004.

[7] S. Aji, S. Fogal, R. McEliece, and B. Wang, "Constrained entropy, free energy, and the legendre transform," 2005. Submitted to International Symposium on Information Theory.

[8] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer-Verlag, 1999.

[9] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley Interscience, 1991.