Forty-Sixth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 23-26, 2008

Asymptotic Weight Enumerators of Randomly Punctured, Expurgated, and Shortened Code Ensembles

Elette C. Boyle and Robert J. McEliece
California Institute of Technology
Pasadena, CA 91125, USA
eboyle@math.mit.edu, rjm@systems.caltech.edu

***Abstract*—In this paper, we examine the effect of random puncturing, expurgating, and shortening on the asymptotic weight enumerator of certain linear code ensembles. We begin by discussing the actions of the three alteration methods on individual codes. We derive expressions for the average resulting code weight enumerator under each alteration. We then extend these results to the spectral shape of linear code ensembles whose original spectral shape is known, and demonstrate our findings on two specific code ensembles: the Shannon ensemble and the regular (j, k) Gallager ensemble.**

I. Introduction

Methods of altering linear binary codes such as puncturing, expurgation, and shortening are important tools for generating good new codes with desired parameters from old codes. A desire for such codes arises, for instance, in data storage, where it is beneficial to construct good codes whose length is a power of 2. Indeed, Sony Corporation currently utilizes shortened BCH codes for error correction in USB data storage devices [9].

Because of these applications, puncturing, expurgation, and shortening are well-studied in the case of finite—typically, short—codes. However, little work has been done to characterize their properties for codes of asymptotically large length n. This topic is of interest since modern iterative decoding methods allow codes of very long length to be used efficiently.

In this paper, we approach the subject of codes of asymptotically large n by studying linear code ensembles.

Definition I.1. *An ensemble of linear codes is a sequence $C_{n_1}, C_{n_2}, \ldots$ of sets of linear codes with common rate R, where C_{n_i} is a set of (n_i, k_i) codes with $k_i/n_i = R$ [1].*

Results proved for code ensembles will imply corresponding results on the existence of codes with certain properties. For a given characteristic we are guaranteed the existence of at least one code within the ensemble—and in many cases, almost all codes—whose characteristics meet or exceed the ensemble average [2].

This work was supported by grants from Sony Corporation, the Caltech Lee Center for Advanced Networking, and NSF Grant No. CCF-0514881

For a linear code $C \subseteq (\mathbb{Z}/2\mathbb{Z})^n$, consider the *weight enumerator* of C—that is, the collection of values $\{A_j\}$, $0 \le j \le n$, such that A_j is equal to the number of codewords in C of weight j. For code ensembles, we consider the *ensemble spectral shape* $r(\delta)$, where $0 \le \delta \le 1$ corresponds to a fraction of n. The ensemble spectral shape describes the asymptotic behavior of the weight enumerator; the term is formally introduced in Section IV.

The weight enumerator and the ensemble spectral shape provide important characterization information for codes in the finite and asymptotic cases, respectively. In the finite case, for instance, the weight enumerator of a binary code dictates the decoder error probability P_E of the code within a bounded-distance decoder model [7]

$$P_E(h) = \frac{1}{\binom{n}{h}} \sum_{s=0}^{t} \sum_{l=h-s}^{h+s} A_l \binom{n-1}{\frac{1}{2}(s+h-l)} \binom{l}{\frac{1}{2}(s-h+l)}.$$

For binary code ensembles, a tight bound depending only on the ensemble spectral shape $r(\delta)$ has been demonstrated for the minimum signal-to-noise ratio at which the codes can reliably transmit information on a Gaussian channel [4],[10]

$$\left(\frac{E_b}{N_0}\right)_{\min} = \frac{1}{R} \max_{0 \le \theta \le (1-R)} \left\{ \left(1 - e^{-2r(\theta)}\right) \frac{1-\theta}{2\theta} \right\}.$$

For any given binary linear code and a fixed value of $p \in \{1, \ldots, n\}$, there exist several choices of p-puncturings, p-expurgations, and p-shortenings. For instance, there exist $\binom{n}{p}$ different p-puncturings. Each choice yields a different weight enumerator in general, dependent on specifics of the code. Instead of attempting to treat all these cases exhaustively, we instead consider the average weight enumerator over all such choices.

We will refer to the action of puncturing, expurgating, or shortening as *altering* a code. In this paper, we derive an expression for the expected spectral shape of a randomly altered linear code ensemble in terms of the original ensemble spectral shape. In this process, we first calculate the expected weight enumerator for randomly altered individual codes. We then utilize our results on code ensembles to study the effect of the three alteration methods on the spectral shape

of two specific ensembles: the Shannon ensemble of random unrestricted linear codes of a common rate, and the regular (j,k) Gallager ensemble. We show randomly altering the Shannon ensemble yields another Shannon ensemble of a particular rate. Randomly altering the regular (j,k) Gallager ensemble by different amounts yields new ensembles with continuously adjustable rates and good distance properties.

II. Puncturing, Expurgating, and Shortening Linear Codes

We begin by introducing and discussing three common code alterations: puncturing, expurgation, and shortening. Each has a different effect on the dimension of the code, k, and the dimension of the ambient space, n. The three alterations can be defined in terms of their action on either the generator or parity-check matrix of the code.

A. Puncturing

Puncturing a code is achieved by removing a collection of p columns from the code generator matrix G. This will decrease the dimension of the ambient vector space from n to $n-p$. We see that removing rows from G will, in effect, delete a fixed set of bits from each codeword in the code [3].

Formally, puncturing a code does not decrease the dimension of the code itself, k. That is, removing the relevant columns from the generator matrix should not decrease the rank of the matrix. In order for this to be true, limitations must clearly be imposed on the number of columns removed from a code. Indeed, puncturing p bits will decrease the dimension of a code if there exist two codewords that vary only within these p bits. Since we are considering the average weight enumerator over all possible p-puncturings, this situation will occur precisely when the minimum distance d_{min} of the code is less than or equal to p.

The inverse of puncturing is *extending* a linear code, which is performed by adding columns to G.

B. Expurgation

A code is *expurgated* by removing p rows of the generator matrix G—or, equivalently, introducing p additional parity constraints by adding p rows to the parity-check matrix H. Since a linear code is formed as the rowspace of its generator matrix, removing rows from G will yield a linear subspace of the original k-dimensional code. Indeed, expurgation decreases the number of codewords contained in a code while leaving the remaining codewords unaffected [3].

For any given (n,k) linear code C, there are several $k \times n$ matrices which generate C. The selection of a single generator matrix representative G corresponds to a choice of basis for the code. However, removing rows of this matrix through expurgation can only generate subspaces spanned by a subset of those particular basis vectors. Unlike the cases of the other two code alterations, the average expurgated weight enumerator is not independent of the choice of G. We demonstrate this with the following constructed example. Let

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad G' = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

be two generator matrices. Note that G and G' share the same rowspace, and thus both represent the same code. Removing row 1 from each matrix yields the same expurgated code, since the two matrices will now be identical. Removing row 3 from each matrix yields the same expurgated code as well, as the rowspaces of the two matrices are still the same. However, removing row 2 yields two different codes with different weight enumerators. This implies the average expurgated weight enumerator would be different for G and G', even though they are generator matrices for the same code.

To avoid this problem, we instead define a p-expurgated code to be any of the $(k-p)$-dimensional linear subspaces of the code. This definition is intrinsically independent of the choice of basis.

The inverse of expurgation is *augmentation*, which can be performed by adding rows to G or removing rows from H. In terms of the more general definition, a p-augmentation of a code C is simply a $(k+p)$-dimensional linear subspace of $(\mathbb{Z}/2\mathbb{Z})^n$ containing C.

C. Shortening

Shortening a code by p bits is achieved by removing p columns from the parity-check matrix H. Shortening decreases both the number of codewords in the code and the length of the words themselves; that is, both k and n are decreased [3]. We can deduce properties of shortened codes by demonstrating the shortening action is a composition of expurgating and puncturing.

Proposition II.1. *Shortening a code by removing p columns $\{i_1,\ldots,i_p\}$ from H is equivalent to expurgating to the subcode of vectors with 0 in bit positions $\{i_1,\ldots,i_p\}$, and then puncturing out these p bits.*

Proof: Let C be the original code and C' be the shortened code. We may assume without loss of generality that $\{i_1,\ldots,i_p\}$ are the last p columns in the parity-check matrix H. Denote the new, smaller parity-check matrix by H'. The result is then equivalent to the statement that a $(n-p)$-vector $x' = (x_1,\ldots,x_{n-p})$ is contained in C' if and only if the corresponding n-vector $x = (x_1,\ldots,x_{n-p},0,\ldots,0)$ is contained in the original code C.

Let $x' \in C'$. This implies $h' \cdot x' = 0$ for each row h' of the matrix H'. Then, for any values of the additional columns in H, we must have $h \cdot x = 0$, where the additional bits of x are all 0 as above. Thus, x is in the null space of H, and so is contained within C.

Suppose $x \in C$ is of the form $(x_1,\ldots,x_{n-p},0,\ldots,0)$, $x_i \in \{0,1\}$. We have that $h \cdot x = 0$ for all rows h of the parity-check matrix H. However, from the form of x, this

TABLE I
SUMMARY OF ALTERATION TYPES AND THEIR EFFECTS ON n, k, G, AND H

	n	k	G	H
Puncture	$n-p$	k	Remove columns	
Expurgate	n	$k-p$	Remove rows	Add rows
Shorten	$n-p$	$k-p$	Remove rows and columns	Remove columns

implies $h' \cdot x' = 0$, where each h' is $(n-p)$-vector composed of the first $(n-p)$ entries in h. Thus, x' is in the null space of H', and so is contained in C'. ∎

Viewed in terms of this composition of expurgation and puncturing, p-shortenings can be described by removing p rows and p columns from an appropriate representative generator matrix G.

We also note that the shortened code C^{short} can be calculated by puncturing the dual code and taking the dual, $(C^{\perp\text{punct}})^{\perp}$. The dual code $C^{\perp}$ of the linear code C is defined by reversing the roles of the generator matrix, G, and the parity-check matrix, H. The punctured dual code is then defined by generator matrix H' (with the relevant columns removed) and parity-check matrix G; the dual of this code is hence defined by generator matrix G and parity-check matrix H'.

$$\begin{array}{ccc} C & \xrightarrow{\text{dual}} & C^{\perp} \\ \downarrow \text{short} & & \downarrow \text{punct} \\ C^{\text{short}} & \xleftarrow{\text{dual}} & C^{\perp\text{punct}} \end{array}$$

The inverse of shortening is *lengthening* a linear code. Lengthening can be achieved by adding rows and columns to G or adding columns to H.

III. RESULTS FOR INDIVIDUAL CODES

In this section we investigate the average weight enumerator of a linear code that has been randomly p-punctured, p-expurgated, or p-shortened.

Theorem III.1. *(Puncture)*
Let C be a (n,k) linear code with weight enumerator A_i, and let $p \in \mathbb{N}$ such that $0 < p < d_{min}$, where d_{min} is the minimum distance of C. Then the average weight enumerator of the punctured code C^{punct}, over all possible p-puncturings, will be

$$\overline{A}_i^{punct} = \sum_{w=0}^{n} \frac{\binom{w}{i}\binom{n-w}{n-p-i}}{\binom{n}{p}} A_w(C).$$

Proof: Consider a word c of weight w in the original code C. In order to p-puncture c to a word of weight i, one must remove $w-i$ ones from the vector and $p-(w-i)$ zeros. There are precisely

$$\binom{w}{w-i}\binom{n-w}{p-(w-i)} = \binom{w}{w-i}\binom{n-w}{n-p-i}$$

such choices of p bits. Note that this value is zero if $i > w$ or $i < w-p$. Since p is less than the minimum distance d_{min} of C, no two codewords in C will puncture to the same codeword. Thus, the total number of weight i codewords in C^{punct} yielded from codewords of original weight w will be $\binom{w}{w-i}\binom{n-w}{n-p-i}A_w$. Hence, the average weight enumerator of the punctured codes is

$$\overline{A}_i^{\text{punct}} = \sum_{w=0}^{n} \frac{\binom{w}{i}\binom{n-w}{n-p-i}}{\binom{n}{p}} A_w(C).$$

∎

Theorem III.2. *(Expurgate)*
Let C be a (n,k) linear code with weight enumerator A_i. Then the average weight enumerator of the p-expurgated codes is

$$\overline{A}_i^{exp} = \frac{2^{k-p}-1}{2^k-1} A_i, \ i \neq 0.$$

Proof: The number of m-dimensional subspaces of the code C is the Gaussian number $\begin{bmatrix} k \\ m \end{bmatrix}_2$. Note the 2 subscript is to denote computations are performed over the field $\mathbb{Z}/2\mathbb{Z}$ [11].

Expurgating a code results in a decreased number of codewords, but those codewords which remain are not changed. Each nonzero codeword $c \in C$ is contained in exactly

$$\begin{bmatrix} k-1 \\ (k-p)-1 \end{bmatrix}_2$$

$(k-p)$-dimensional subspaces. Thus, for each i, there will be an average of

$$\begin{aligned} &\left(\begin{bmatrix} k-1 \\ (k-p)-1 \end{bmatrix}_2 \Big/ \begin{bmatrix} k \\ (k-p) \end{bmatrix}_2 \right) A_i = \\ &= \frac{(2^{k-1}-1)\cdots(2^{p+1}-1)}{(2^{k-p-1}-1)\cdots(2^1-1)} \frac{(2^{k-p}-1)\cdots(2^1-1)}{(2^k-1)\cdots(2^{p+1}-1)} A_i \\ &= \frac{2^{k-p}-1}{2^k-1} A_i \end{aligned}$$

words of weight i contained in the p-expurgated subcode. ∎

We note that the expression for the expurgated weight enumerator involves multiplication by a constant independent of the weight i.

Theorem III.3. *(Shorten)*
Let C be a (n,k) linear code with weight enumerator A_i. Then the average weight enumerator of the p-shortened codes is

$$\overline{A}_i^{short} = \frac{\binom{n-i}{p}}{\binom{n}{p}} A_i, \ i \neq 0.$$

Proof: In Proposition II.1, we showed that a p-shortened code is formed by expurgating to the subcode of vectors with zeros in the relevant p bits, and then puncturing these bits. The puncturing action will not affect the weight enumerator, as we are only deleting bits of zero weight. It remains to calculate, then, the effect of the code expurgation.

A codeword $c \in C$ will appear in the expurgated code if and only if it has a 0 bit in the corresponding p bit positions. A codeword of weight i will satisfy this requirement for precisely $\binom{n-i}{p}$ of the $\binom{n}{p}$ possible bit choices. The result follows. ∎

IV. Modifying Linear Code Ensembles

We now extend our results on altered weight enumerators from individual codes to ensembles of linear codes. We investigate the asymptotic behavior of the weight enumerator of the codes when punctured, expurgated, or shortened by a constant ratio αn of the overall codeword length n.

Definition IV.1. (1) *An ensemble of linear codes is a sequence $C_{n_1}, C_{n_2}, \ldots$ of sets of linear codes with common rate R, where C_{n_i} is a set of (n_i, k_i) codes with $k_i/n_i = R$.*
(2) *$A_0(C), \ldots, A_{n_i}(C)$ is the weight enumerator for the code $C \in C_{n_i}$*
(3) *$\overline{A}_0^{(n)}, \ldots, \overline{A}_n^{(n)}$ is the average weight enumerator for C_n,*

$$\overline{A}_h^{(n)} := \frac{1}{|C_n|} \sum_{C \in C_n} A_h(C), \text{ for } h = 0, 1, \ldots, n \quad [I].$$

In code ensembles, the value of n tends to infinity in the sequence. As it does so, the terms of the weight enumerator will approach infinity as well. We are interested in studying how quickly the weight enumerator grows. We do so by considering the *ensemble spectral shape*, which we now define.

Definition IV.2. (1) *For a code ensemble with weight enumerator A_j, $n \in \mathbb{N}$, we define for $\delta \in [0, 1]$,*

$$r_n(\delta) := \frac{1}{n} \log \overline{A}_{\lfloor \delta n \rfloor}^{(n)}.$$

(2) *The ensemble spectral shape of a linear code ensemble is defined for $\delta \in [0, 1]$ to be the limit*

$$r(\delta) := \lim_{n \to \infty} r_n(\delta).^{\dagger}$$

We now characterize the expected ensemble spectral shape of a randomly altered code ensemble in terms of the spectral shape of the original ensemble.

Theorem IV.3. *(Punctured Code Ensemble) Let $r(\delta)$ be the spectral shape of a linear code ensemble with positive fractional minimum distance δ_0. Then randomly puncturing the ensemble by $p = \alpha n$, $0 < \alpha < \delta_0$, yields an ensemble with expected spectral shape*

$$r^{punct}(\delta) = \frac{1}{c}\left[\max_{0 \le \lambda \le 1} \left\{ \lambda H\left(\frac{c\delta}{\lambda}\right) + (1-\lambda) H\left(\frac{\alpha + c\delta - \lambda}{1-\lambda}\right) + r(\lambda) \right\} - H(\alpha) \right]$$

where

$$H(x) := -x \log x - (1-x) \log(1-x)$$

is the entropy function and $c = (1 - \alpha)$.

Sketch of Proof. In Theorem III.1 we showed puncturing a single code C by $p = \alpha n$ yields

$$\overline{A}_i^{\text{punct}} = \sum_{w=0}^{n} \frac{\binom{w}{i}\binom{n-w}{\alpha n - (w-i)}}{\binom{n}{\alpha n}} A_w(C).$$

For each n, we define the function

$$\tilde{r}_n(\delta) = \frac{1}{n} \log\left(\frac{1}{|C_n|} \sum_{C \in C_n} \left(\sum_{w=0}^{n} \frac{\binom{w}{\lfloor \delta n \rfloor}\binom{n-w}{\alpha n - (w - \lfloor \delta n \rfloor)}}{\binom{n}{\alpha n}} A_w(C) \right) \right).$$

Taking the limit of this function as n tends to infinity yields the ensemble spectral shape, $\tilde{r}(\delta)$. The only term in the expression of $\tilde{r}(\delta)$ that depends on the index $C \in C_n$ is the weight enumerator term $A_w(C)$. We may thus pull the exterior sum inside, to yield

$$\begin{aligned} \tilde{r}(\delta) &= \lim_{n \to \infty} \frac{1}{n} \log\Bigg(\sum_{w=0}^{n} \frac{\binom{w}{\lfloor \delta n \rfloor}\binom{n-w}{\alpha n - (w - \lfloor \delta n \rfloor)}}{\binom{n}{\alpha n}} \left(\frac{1}{|C_n|} \sum_{C \in C_n} A_w(C) \right) \Bigg) \\ &= \lim_{n \to \infty} \frac{1}{n} \log\left(\sum_{w=0}^{n} \frac{\binom{w}{\lfloor \delta n \rfloor}\binom{n-w}{\alpha n - (w - \lfloor \delta n \rfloor)}}{\binom{n}{\alpha n}} \overline{A}_w^{(|C_n|)} \right). \end{aligned}$$

Express w as a fraction λn of n, where $0 \le \lambda \le 1$. Utilizing Stirling's Approximation gives $\binom{a}{ca} \to e^{aH(c)}$ for each binomial coefficient. Thus, in the limit we have

$$\tilde{r}(\delta) = \lim_{n \to \infty} \frac{1}{n} \log\left(\sum_{\lambda n = 0}^{n} \frac{e^{\lambda n H(\delta/\lambda)} e^{n(1-\lambda) H\left(\frac{\alpha + \delta - \lambda}{1-\lambda}\right)}}{e^{nH(\alpha)}} \overline{A}_{\lambda n}^{(|C_n|)} \right).$$

Now, the number of terms in the above sum will increase linearly in n, while the size of each term will grow exponentially in n. In the limit, the total value of the sum will go as the value of its maximal term. Since the logarithm function and multiplication by $\frac{1}{n}$ are monotonically increasing functions, they can be brought inside the maximization expression,

[†]Technically, there may exist situations where this limit does not exist. In these situations, one can formally discuss the spectral shape in terms of lim sup, and everything will follow through in a similar fashion.

yielding

$$\begin{aligned}\tilde{r}(\delta) &= \lim_{n\to\infty} \max_{0\le\lambda\le 1} \left\{ \lambda H\left(\frac{\delta}{\lambda}\right) + (1-\lambda) H\left(\frac{\alpha+\delta-\lambda}{1-\lambda}\right) \right. \\ &\qquad \left. - H(\alpha) + \frac{1}{n}\log\left(\overline{A}_{\lambda n}^{(|C_n|)}\right)\right\} \\ &= \max_{0\le\lambda\le 1} \left\{ \lambda H\left(\frac{\delta}{\lambda}\right) + (1-\lambda) H\left(\frac{\alpha+\delta-\lambda}{1-\lambda}\right) + r(\lambda)\right\} \\ &\qquad - H(\alpha).\end{aligned}$$

Now, this expression is still in terms of the original value of the codeword length, n. However, we wish to express the asymptotic weight enumerator in terms of the post-punctured codeword length, $n(1-\alpha)$. To achieve this, we must introduce a factor of $(1-\alpha)$ to each δ term and scale the overall expression by $\frac{1}{1-\alpha}$. Hence, we have our result,

$$\begin{aligned}r^{\text{punct}}(\delta) = \frac{1}{1-\alpha}\Bigg[&\max_{0\le\lambda\le 1} \left\{ \lambda H\left(\frac{(1-\alpha)\delta}{\lambda}\right) \right. \\ &\left. + (1-\lambda) H\left(\frac{\alpha+(1-\alpha)\delta-\lambda}{1-\lambda}\right) + r(\lambda)\right\} - H(\alpha)\Bigg].\end{aligned}$$

■

Theorem IV.4. *(Expurgated Code Ensemble) Let $r(\delta)$ be the spectral shape of a linear code ensemble with rate R. Then randomly expurgating the ensemble by αn, $0 < \alpha < R$, yields an ensemble with expected spectral shape*

$$r^{exp}(\delta) = -\alpha\log 2 + r(\delta).$$

Proof: By Theorem III.2, expurgating an individual code C by $p = \alpha n$ results in the average weight enumerator

$$\overline{A}_i^{\text{exp}} = \frac{2^{k-\alpha n}-1}{2^k-1} A_i,\ i \neq 0.$$

Thus, we have

$$\begin{aligned}r^{\text{exp}}(\delta) &= \lim_{n\to\infty}\frac{1}{n}\log\left(\frac{1}{|C_n|}\sum_{C\in C_n}\frac{2^{Rn-\alpha n}-1}{2^{Rn}-1}A_{\lfloor\delta n\rfloor}(C)\right) \\ &= \lim_{n\to\infty}\frac{1}{n}\log\left(\frac{2^{Rn-\alpha n}-1}{2^{Rn}-1}\right) \\ &\qquad + \lim_{n\to\infty}\frac{1}{n}\log\frac{1}{|C_n|}\sum_{C\in C_n} A_{\lfloor\delta n\rfloor}(C) \\ &= -\alpha\log 2 + r(\delta).\end{aligned}$$

■

Theorem IV.5. *(Shortened Code Ensemble) Let $r(\delta)$ be the spectral shape of a linear code ensemble with positive fractional minimum distance δ_0. Then randomly shortening the ensemble by αn, $0 < \alpha < \delta_0$, yields an ensemble with expected spectral shape*

$$r^{short}(\delta) = \frac{1}{c}\left[(1-c\delta)\ H\left(\frac{\alpha}{1-c\delta}\right) - H(\alpha) + r(c\delta)\right],$$

where $c = (1-\alpha)$.

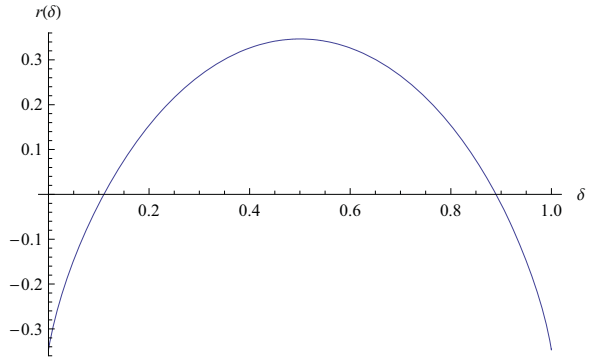

Fig. 1. Ensemble spectral shape for the rate $\frac{1}{2}$ Shannon ensemble.

Proof: By Theorem II.1, shortening an individual code C by $p = \alpha n$ yields the average weight enumerator

$$\overline{A}_i^{\text{short}} = \frac{\binom{n-i}{\alpha n}}{\binom{n}{\alpha n}} A_i,\ i \neq 0.$$

This implies an ensemble spectral shape of

$$\tilde{r}(\delta) = \lim_{n\to\infty}\frac{1}{n}\log\left(\frac{1}{|C_n|}\sum_{C\in C_n}\frac{\binom{n-\lfloor\delta n\rfloor}{\alpha n}}{\binom{n}{\alpha n}}A_{\lfloor\delta n\rfloor}(C)\right).$$

Stirling's Approximation gives that $\binom{a}{ca} \to e^{aH(c)}$ as n tends to infinity. Utilizing this approximation gives

$$\begin{aligned}\tilde{r}(\delta) &= \lim_{n\to\infty}\frac{1}{n}\log\left(\frac{e^{n(1-\delta)H\left(\frac{\alpha}{1-\delta}\right)}}{e^{nH(\alpha)}}\frac{1}{|C_n|}\sum_{C\in C_n}A_{\lfloor\delta n\rfloor}(C)\right) \\ &= \lim_{n\to\infty}\left[(1-\delta)H\left(\frac{\alpha}{1-\delta}\right)\right. \\ &\qquad \left. - H(\alpha) + \frac{1}{n}\log\overline{A}_{\lfloor\delta n\rfloor}(C)\right] \\ &= (1-\delta)H\left(\frac{\alpha}{1-\delta}\right) - H(\alpha) + r(\delta).\end{aligned}$$

We again have an expression in terms of the original codeword length n instead of the desired post-punctured codeword length, $n(1-\alpha)$. To make this change, we must again introduce a factor of $(1-\alpha)$ to each δ and scale the overall expression by $\frac{1}{1-\alpha}$, yielding

$$\begin{aligned}r^{\text{short}}(\delta) = \frac{1}{1-\alpha}\Bigg[&\big(1-(1-\alpha)\delta\big)H\left(\frac{\alpha}{1-(1-\alpha)\delta}\right) \\ &- H(\alpha) + r\big((1-\alpha)\delta\big)\Bigg].\end{aligned}$$

■

V. Results on Specific Code Ensembles

We now apply our results from the previous section to two common code ensembles: the Shannon ensemble and the regular (j,k) Gallager ensemble.

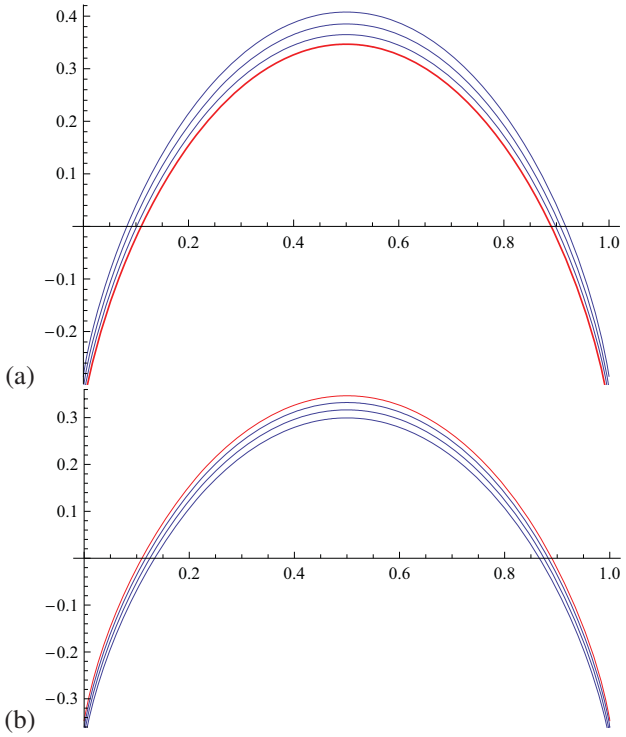

Fig. 2. Ensemble spectral shape for the rate $\frac{1}{2}$ Shannon ensemble (red, thick) in addition to the spectral shapes for (a) the punctured ensembles with $\alpha = 0.04, 0.08, 0.12$, and (b) the shortened ensembles with $\alpha = 0.04, 0.08, 0.12$.

A. Shannon Ensemble

The rate R Shannon ensemble is composed of the random linear codes of rate R. The Shannon ensemble is characterized by the average weight enumerator

$$\overline{A}_h^{(n)} = \binom{n}{h} 2^{-n(1-R)}.$$

This yields an ensemble spectral shape of

$$r_S(\delta) = H(\delta) - (1-R)\log 2 \quad [1].$$

The spectral shape for the rate $\frac{1}{2}$ Shannon ensemble is shown in Figure 1.

Corollary V.1. *(1) Puncturing the rate R Shannon ensemble by $p = \alpha n$ yields an ensemble with expected spectral shape equal to that of the Shannon ensemble of rate $R' = \frac{R}{1-\alpha}$.*

(2) Expurgating the rate R Shannon ensemble by $p = \alpha n$ yields an ensemble with expected spectral shape equal to that of the Shannon ensemble of rate $R'' = R - \alpha$.

(3) Shortening the rate R Shannon ensemble by $p = \alpha n$ yields an ensemble with expected spectral shape equal to that of the Shannon ensemble of rate $R''' = \frac{R-\alpha}{1-\alpha}$.

Proof: (1) From Theorem IV.3, we know that puncturing a linear code ensemble by αn takes the ensemble spectral shape from $r_S(\delta)$ to

$$r_S^{\text{punct}}(\delta) = \frac{1}{c}\left[\max_{0\le\lambda\le 1}\left\{\lambda H\left(\frac{c\delta}{\lambda}\right) + (1-\lambda)H\left(\frac{\alpha + c\delta - \lambda}{1-\lambda}\right) + r_S(\lambda)\right\} - H(\alpha)\right],$$

where $c = (1-\alpha)$. For the Shannon ensemble of rate R, we have $r_S(\delta) = H(\delta) - (1-R)\log 2$. In this case, the expression to be maximized is a smooth function of λ and can thus be maximized using calculus. Indeed, the partial derivative with respect to λ is equal to 0 when $\lambda = \delta + \alpha/2$. A quick investigation of the function behavior around the critical point and at the endpoints $\lambda = 0, 1$ shows this solution to be the global maximum. Plugging in $\lambda = \delta + \alpha/2$ yields

$$\begin{aligned} r_S^{\text{punct}}(\delta) =& \frac{1}{c}[-c\delta\log(c\delta) - (1-\alpha-c\delta)\log(1-\alpha-\delta) \\ &+ (1-\alpha)\log(1-\alpha) - (1-(R+\alpha))\log(2)] \\ =& H(\delta) - \left(1 - \frac{R}{1-\alpha}\right)\log(2), \end{aligned}$$

which is equivalent to the spectral shape of the Shannon ensemble of rate $\frac{R}{1-\alpha}$.

(2) By Theorem IV.4, expurgating the Shannon ensemble by αn will result in the ensemble spectral shape

$$\begin{aligned} r_S^{\text{exp}}(\delta) &= r_S(\delta) - \alpha\log 2 \\ &= H(\delta) - (1-R)\log 2 - \alpha\log 2 \\ &= H(\delta) - (1-(R-\alpha))\log 2. \end{aligned}$$

This is precisely the spectral shape of the Shannon ensemble of rate $R - \alpha$.

(3) Theorem IV.5 gives that the αn-shortened Shannon ensemble will have an ensemble spectral shape of

$$r^{\text{short}}(\delta) = \frac{1}{c}\left[(1-c\delta)\ H\left(\frac{\alpha}{1-c\delta}\right) - H(\alpha) + H(c\delta) - (1-R)\log 2\right],$$

where $c = (1-\alpha)$. After substituting $H(x) = -x\log x - (1-x)\log(1-x)$, algebraic manipulation yields

$$\begin{aligned} r^{\text{short}}(\delta) &= \frac{1}{1-\alpha}\Big[-(1-\alpha)(1-\delta)\log(1-\delta) \\ &\qquad - (1-\alpha)\delta\log\delta - (1-R)\log 2\Big] \\ &= -(1-\delta)\log(1-\delta) - \delta\log\delta - \frac{1-R}{1-\alpha}\log 2 \\ &= H(\delta) - \left(1 - \frac{R-\alpha}{1-\alpha}\right)\log 2 \end{aligned}$$

■

These results are as expected, and serve as a check for our calculations. For a given rate R, the Shannon ensemble shares its spectral shape with the ensemble of codes of $Rn_i \times n_i$ binary generator matrices whose entries are chosen randomly

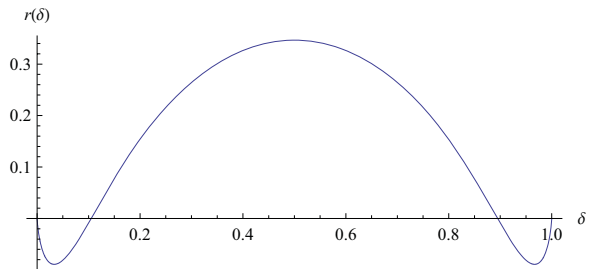

Fig. 3. Ensemble spectral shape for the regular $(8, 16)$ Gallager ensemble.

and independently with equal probability. In this case, we see clearly that deleting αn_i columns from the generator matrices by puncturing will simply leave generator matrices of size $Rn_i \times (1-\alpha)n_i$ with random entries. That is, we are left with an identical ensemble of codes with rate

$$R' = \frac{Rn_i}{(1-\alpha)n_i} = \frac{R}{1-\alpha},$$

which thus possesses an ensemble spectral shape as described above. Similarly, deleting αn_i rows from the generator matrix by expurgating will yield an identical ensemble of codes with rate

$$R'' = \frac{Rn_i - \alpha n_i}{n_i} = R - \alpha,$$

and deleting αn_i rows and columns from G by shortening will yield an identical ensemble with rate

$$R''' = \frac{Rn_i - \alpha n_i}{n_i - \alpha n_i} = \frac{R-\alpha}{1-\alpha}.$$

Figure 2(a) depicts the spectral shape of the rate $\frac{1}{2}$ Shannon ensemble along with the spectral shapes for three resulting punctured ensembles, with $\alpha = 0.04, 0.08$, and 0.12. From Corollary $V.1$, it follows that these correspond to the Shannon ensembles of rate $0.521, 0.543$, and 0.568. The ensembles with higher rates correspond to the higher spectral shapes in the plot. Figure 2(b) shows the spectral shape of the rate $\frac{1}{2}$ Shannon ensemble against the corresponding shortened ensembles with $\alpha = 0.04, 0.08$, and 0.12. It can similarly be shown that these ensembles exhibit the spectral shape of Shannon ensembles of rate $0.479, 0.457$, and 0.432.

B. Regular (j, k) Gallager Ensemble

The regular (j, k) Gallager ensemble is composed of codes whose parity check matrices have exactly j ones in each column and k ones in each row. The spectral shape of the regular (j, k) Gallager code ensemble can be expressed parametrically as

$$\delta_{j,k}(s) = \frac{1}{k}\frac{\partial \mu}{\partial s}(s,k)$$
$$r_{j,k}(s) = \frac{j}{k}\left(\mu(s,k) - s\frac{\partial \mu}{\partial s}(s,k) + (k-1)\log 2\right) - (j-1)H\left(\frac{1}{k}\frac{\partial \mu}{\partial s}(s,k)\right),$$

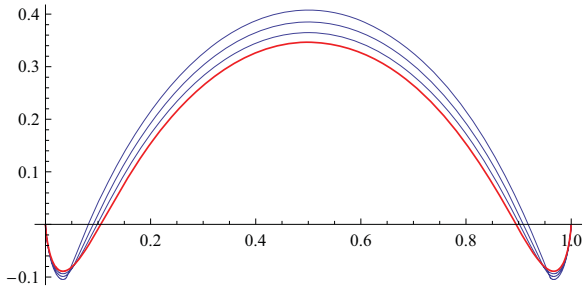

Fig. 4. Ensemble spectral shape for the regular $(8, 16)$ Gallager ensemble (red, thick) in addition to the spectral shapes for the punctured ensembles with $\alpha = 0.04, 0.08, 0.12$.

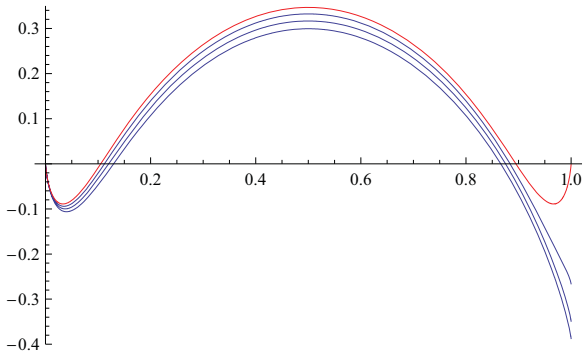

Fig. 5. Ensemble spectral shape for the regular $(8, 16)$ Gallager ensemble (red, thick) along with the spectral shapes for the shortened ensembles with $\alpha = 0.04, 0.08, 0.12$.

where s ranges over all $\mathbb{R}$ and $\mu(s, k)$ is given by

$$\mu(s,k) := \log \frac{(1+e^s)^k + (1-e^s)^k}{2^k}. \quad [1]$$

The spectral shape of the regular $(8, 16)$ Gallager ensemble is exhibited in Figure 3.

Note that the regular Gallager ensemble is a generalization of the classical Gallager ensemble of Low-Density Parity Check (LDPC) codes, introduced by Robert Gallager in 1963 [5]. It has been demonstrated that both ensembles share the same spectral shape [6]. These ensembles are also very closely related to the ensemble of LDPC codes treated by David MacKay in [8].

By calculating an explicit expression for $r_{j,k}(\delta)$ and utilizing the results from the previous section, we were able to generate a parametric expression for the spectral shape of the punctured regular Gallager ensemble. Figure 4 depicts the spectral shape of the regular $(8, 16)$ Gallager ensemble against those of three of its punctured ensembles, with puncturing parameters $\alpha = 0.04, 0.08$, and 0.12. The ensembles with greater puncturing parameters have spectral shapes with a higher maximum value and a smaller minimum intercept. In a similar fashion, we can use Theorem IV.5 to produce an expression for the spectral shape of the regular (j, k) ensemble shortened by αn. An example of the shortened $(8, 16)$ ensemble is shown in Figure 5. Note that shortening the ensemble in this case yields a spectral shape that is

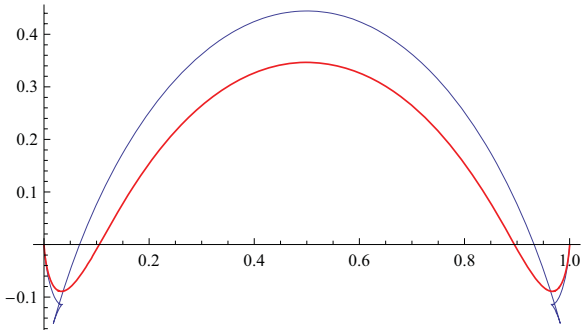

Fig. 6. The spectral shape for the punctured regular $(8, 16)$ Gallager ensemble with $\alpha = 0.22$. The calculated parametric expression exhibits erroneous loops for large values of α.

asymmetric; this is logical, as the parity-check matrices of the modified ensemble codes will now contain rows of odd weight.

When the puncturing parameter α for the regular $(8, 16)$ Gallager ensemble is set to be greater than the approximate ensemble fractional minimum distance $\delta \approx 0.105$, the derived parametric relation breaks down. For large values of α, the expression for the independent variable δ is no longer monotonically increasing, and hook-like loops occur in the parametric plot. Such loops are depicted in Figure 6. A similar issue appears when the regular $(3, 6)$ Gallager ensemble is punctured by $\alpha \geq 0.365$. It is interesting to note that the problem does not occur in this case for values of α between 0.365 and the approximate ensemble fractional minimum distance $\delta \approx 0.025$. Further investigation will be required to assess the mathematical cause for these phenomena.

VI. Conclusion

We have successfully expressed the expected spectral shape of a randomly altered linear code ensemble in terms of the spectral shape of the original ensemble. In this process, we derived the average weight enumerator of randomly altered individual codes. We used our results to characterize altered Shannon ensembles and to calculate the spectral shape of altered regular (j, k) Gallager ensembles. Interesting problems for future work include applying the general results to additional ensembles, investigating the limitations of alterations and determining the cause for the observed breakdowns, and analyzing the connection between the calculated ensemble spectral shapes and the "goodness" of the codes.

References

[1] Aji, Srinivas, et al. "BSC Thresholds for Code Ensembles Based on 'Typical Pairs' Decoding." *Codes, Systems, and Graphical Models*. Ed. Brian Marcus, Joachim Rosenthal. New York: Springer, 2001.

[2] Alon, Noga, and Spencer, Joel H. *The Probabilistic Method*. New York: Wiley-Interscience, 2000.

[3] Berlekamp, Elwyn R. *Algebraic Coding Theory*. New York: McGraw-Hill Book Company, 1968.

[4] Divsalar, D. "A Simple Tight Bound on Error Probability of Block Codes with Application to Turbo Codes. *TMO Progress Report 42-139*, November 1999.

[5] Gallager, Robert G. *Low Density Parity Check Codes*. M.I.T. Press, 1963.

[6] Litsyn, Simon, and Shevelev, Vladimir. "On Ensembles of Low-Density Parity-Check Codes: Asymptotic Distance Distributions." *IEEE Transactions on Information Theory*. Vol 48, No 4, April 2002.

[7] Kim, Min-Goo. "Undetected Error Probabilities of Binary Primitive BCH Codes for Both Error Correction and Detection." *IEEE Transactions on Communications*. Vol 44, No 5, May 1996.

[8] MacKay, David J.C. "Good Error-Correcting Codes based on Very Sparse Matrices." *IEEE Transactions on Information Theory*. Vol 45, No 2, Jan. 1999.

[9] Shinbashi, T. Sony Corporation. Personal communication. August 2007.

[10] Sweatlock, Sarah. "Asymptotic Weight Analysis of Low-Density Parity-Check (LDPC) Code Ensembles." Dissertation. California Institute of Technology, 2008.

[11] Van Lint, J.H. and Wilson, R.M. *A Course in Combinatorics*. Cambridge: Cambridge University Press, 2001.