Extrinsic Information Transfer Functions, Information Functions, Support Weights, and Duality

A. Ashikhmin, G. Kramer, and S. ten Brink

Bell Laboratories, 600 Mountain Ave, Murray Hill, NJ 07974, USA
E-mail: aea@bell-labs.com, gkr@bell-labs.com, tenbrink@ieee.org

Abstract: *Extrinsic information transfer (EXIT) charts are a tool for predicting the convergence behavior of iterative decoding of concatenated codes. Erasure channel properties are proved that relate a code's EXIT function to its information functions, and thereby to the support weights of its subcodes. The relation is via a refinement of information functions called split information functions, and via a refinement of support weights called split support weights. Split information functions are used to prove another property that relates the EXIT function of a linear code to the EXIT function of its dual.*

Keywords: Extrinsic information transfer charts, information functions, support weights, duality.

1. Introduction

Density evolution was suggested in [1, p. 48] as a tool for predicting the convergence behavior of low–density parity–check (LDPC) codes. The analysis is particularly simple for the binary erasure channel (BEC) because one must compute only the fraction of erasures being passed from one component decoder to another. For example, this is done in [2], [3] for irregular LDPC codes, and in [4] for repeat–accumulate (RA) codes.

We will also restrict attention to erasure channels but consider general encoding schemes. Our analysis tool will be extrinsic information transfer (EXIT) charts [5], and we prove several properties of EXIT functions for the decoding model described in [6]. One property is that EXIT functions can be expressed in terms of what we call *split information functions* and *split weight enumerators*. The former are refinements of the *information functions of a code* introduced in [7], while the latter are refinements of the weight enumerators of a code [8]–[9]. Split information functions are used to relate the EXIT function of a linear code to the EXIT function of its dual.

This paper is organized as follows. In Section 2. we describe the decoding model and EXIT functions. In Section 3. we derive several properties of EXIT functions when the *a priori* information is modeled as coming from a BEC. Section 4. summarizes our results.

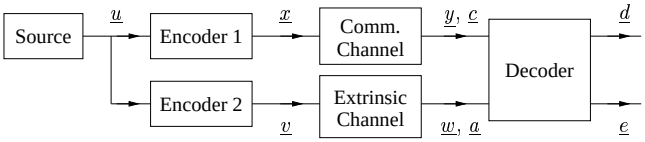

Figure 1: A decoding model.

2. Decoding Model and EXIT

We use the decoding model shown in Fig. 1 (see [6]). A source produces a vector $\underline{u}$ of k independent information bits each taking on the values 0 and 1 with probability 1/2. Encoder 1 maps $\underline{u}$ to a binary length n code word $\underline{x}$, and Encoder 2 maps $\underline{u}$ to a binary length m code word $\underline{v}$. The decoder receives two vectors: a noisy version $\underline{y}$ of $\underline{x}$ and a noisy version $\underline{w}$ of $\underline{v}$. We call the $\underline{x}$ to $\underline{y}$ channel the *communication channel*, and the $\underline{v}$ to $\underline{w}$ channel the *extrinsic channel*. We will assume that both channels are BECs, and that the communication and extrinsic channel erasure probabilities are q and p, respectively.

The decoder uses $\underline{y}$ and $\underline{w}$ to compute two estimates of $\underline{v}$: the *a posteriori* values $\underline{d}$ and the *extrinsic* values $\underline{e}$. The symbol w_i gives *a priori* information about the V_i with L-values

$$a_i = \log \frac{P(w_i \,|\, V_i = 0)}{P(w_i \,|\, V_i = 1)}. \tag{1}$$

Similarly, the symbol y_j gives *a priori* information about X_j with L-value

$$c_j = \log \frac{P(y_j \,|\, X_j = 0)}{P(y_j \,|\, X_j = 1)}. \tag{2}$$

The *a posteriori* probability (APP) bit decoder computes the L-values

$$d_i = \log \frac{\Pr(V_i = 0 \,|\, \underline{y}, \underline{w})}{\Pr(V_i = 1 \,|\, \underline{y}, \underline{w})}. \tag{3}$$

For further analysis, we write $\underline{v}_{[i]}$ for the vector $\underline{v}_{[i]} = [v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_m]$. We have (see [6])

$$d_i = a_i + e_i \tag{4}$$

where

$$e_i = \log \frac{\Pr(V_i = 0 \,|\, \underline{y}, \underline{w}_{[i]})}{\Pr(V_i = 1 \,|\, \underline{y}, \underline{w}_{[i]})}. \tag{5}$$

The number e_i is called the *extrinsic value* about v_i, and we will consider e_i to be a realization of the random variable E_i.

We define two quantities, namely

$$I_A := \frac{1}{m}\sum_{i=1}^{m} I(V_i; A_i) \tag{6}$$

$$I_E := \frac{1}{m}\sum_{i=1}^{m} I(V_i; E_i)\,. \tag{7}$$

The value I_A is called the *average a priori information* going into the encoder, and I_E is called the *average extrinsic information* coming out of the decoder. An EXIT chart plots I_E as a function of I_A.

Consider first I_A for which we usually have

$$I_A = I(V_1; A_1) = 1 - p. \tag{8}$$

The first equality holds if all V_i have the same distribution, while the second holds if this distribution is the uniform one. We will consider only such codes. Consider next I_E for which we have (see [6])

$$I(V_i; E_i) = I(V_i; \underline{Y}\,\underline{A}_{[i]}) \tag{9}$$

Using this proposition, we have

$$I_E = \frac{1}{m}\sum_{i=1}^{m} I\left(V_i; \underline{Y}\,\underline{A}_{[i]}\right). \tag{10}$$

3. Properties

3.1. Information Functions of a Code

The *information function in h positions* of a code $\mathcal{C}$ was defined in [7] as the average amount of information in h positions of $\mathcal{C}$. More precisely, let n be the code length and $\mathcal{S}_h$ be the set of all subsets of $\{1, 2, \ldots, n\}$ of size h. Let $\mathcal{S} \in \mathcal{S}_h$ with $\mathcal{S} = \{i_1, i_2, \ldots, i_h\}$. We write

$$\underline{x}_{\mathcal{S}} = [\, x_{i_1}, x_{i_2}, \ldots, x_{i_h}\,]$$
$$\mathcal{C}_{\mathcal{S}} = \{\underline{x}_{\mathcal{S}} : \underline{x} \in \mathcal{C}\}\,.$$

Let $\mathcal{C}$ be a linear code and $k_{\mathcal{S}}$ the dimension of $\mathcal{C}_{\mathcal{S}}$. The information function in h positions of $\mathcal{C}$ is

$$e_h = \frac{1}{\binom{n}{h}}\sum_{\mathcal{S}\in\mathcal{S}_h} k_{\mathcal{S}}. \tag{11}$$

We write the unnormalized version of e_h as

$$\tilde{e}_h = \sum_{\mathcal{S}\in\mathcal{S}_h} k_{\mathcal{S}}. \tag{12}$$

We remark that these definitions and the following theory can be extended to nonlinear codes (cf. [7]).

Consider the following simple generalization of e_h. Let $\mathcal{C}$ be the code formed by all pairs $(\underline{v}, \underline{x})$ in Fig. 1. Suppose Encoders 1 and 2 are linear, and that $\mathcal{C}$ is a linear $[m+n, k]$ code. Let $\mathcal{S}_{g,h}$ be the set of all subsets of $\{1, 2, \ldots, m+n\}$ of the form $\{i_1, i_2, \ldots, i_g, j_1, j_2, \ldots, j_h\}$ where $1 \le i_1 < i_2 < \ldots i_g \le m$ and $m+1 \le j_1 < j_2 < \ldots j_h \le m+n$. In other words, $\mathcal{S}_{g,h}$ is the set of subsets of g positions from the first m positions of $\mathcal{C}$, and h positions from the last n positions of $\mathcal{C}$. We define the *split information function in (g,h) positions* of $\mathcal{C}$ as

$$e_{g,h} = \frac{1}{\binom{m}{g}}\frac{1}{\binom{n}{h}}\sum_{\mathcal{S}\in\mathcal{S}_{g,h}} k_{\mathcal{S}}. \tag{13}$$

We write the unnormalized version of $e_{g,h}$ as

$$\tilde{e}_{g,h} = \sum_{\mathcal{S}\in\mathcal{S}_{g,h}} k_{\mathcal{S}}. \tag{14}$$

Note that $\tilde{e}_{0,h} = \tilde{e}'_h$ where $\tilde{e}'_h$ is the information function computed for Encoder 1. Similarly, we have $\tilde{e}_{g,0} = \tilde{e}'_g$ where $\tilde{e}'_g$ is the information function computed for Encoder 2.

The following theorem shows that EXIT functions on a BEC can be computed from the split information functions.

Theorem 1 *If the extrinsic and communication channels are BECs with respective erasure probabilities p and q, and if Encoders 1 and 2 are linear with no idle components, then we have*

$$I_E(p,q) = 1 - \frac{1}{m}\sum_{h=0}^{n}(1-q)^h q^{n-h}\sum_{g=1}^{m}(1-p)^{g-1}p^{m-g} \left[g\cdot\tilde{e}_{g,h} - (m-g+1)\cdot\tilde{e}_{g-1,h}\right]. \tag{15}$$

3.2. Support Weights

The information functions of a code are known to be related to the *support weights* of its subcodes [7]. The support weight $w(\mathcal{C})$ of a code $\mathcal{C}$ is the number of positions where *not* all codewords of $\mathcal{C}$ are zero. The *rth support weight* A_i^r of $\mathcal{C}$ is the number of unique subspaces of $\mathcal{C}$ of dimension r and support weight i. Note that $A_0^0, A_1^1, A_2^1, A_3^1, \ldots, A_n^1$ is the usual weight distribution of a code. Note also that $A_i^r = 0$ for $i < r$. We write $A_i^r = 0$ for $r < 0$.

It is known that $\tilde{e}_h$ can be written in terms of the A_i^r as follows (see [7, Thm. 5]):

$$\tilde{e}_h = \sum_{r=1}^{h} r \sum_{s=0}^{r}(-1)^s\, 2^{\binom{s}{2}} \begin{bmatrix} k-r+s \\ s \end{bmatrix} \sum_{i=0}^{n} \binom{n-i}{h} A_i^{k-r+s} \tag{16}$$

where for all i we define $\binom{i}{0} = 1$ and $\begin{bmatrix} i \\ 0 \end{bmatrix} = 1$, and for $j > 0$ we write

$$\binom{i}{j} = \prod_{\ell=0}^{j-1}\frac{i-\ell}{j-\ell}\,, \qquad \begin{bmatrix} i \\ j \end{bmatrix} = \prod_{\ell=0}^{j-1}\frac{2^i - 2^\ell}{2^j - 2^\ell}.$$

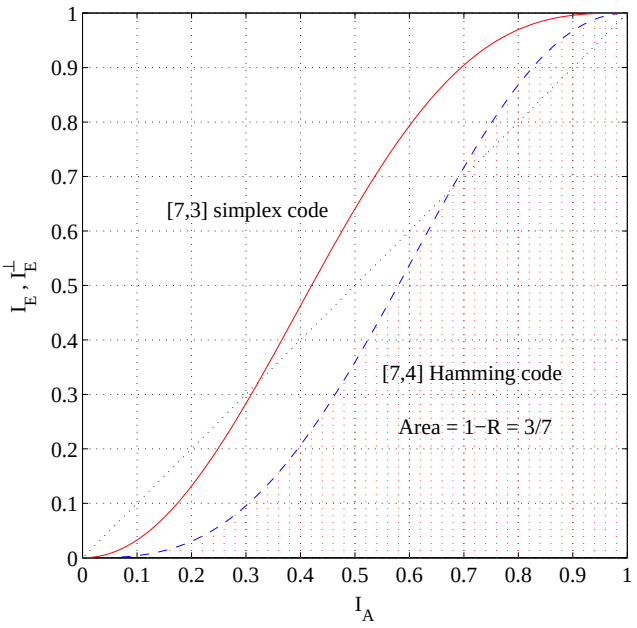

Figure 2: EXIT functions for the $[7, 3]$ simplex code and its dual.

The A_i^r have been determined for some codes. For instance, the $[2^k - 1, k]$ simplex code has (see [7, Sec. IV])

$$A_i^r = \begin{cases} \begin{bmatrix} k \\ r \end{bmatrix} & \text{if } i = 2^k - 2^{k-r} \text{ for } 0 \le r \le k \\ 0 & \text{else.} \end{cases} \tag{17}$$

Inserting (17) into (16), and performing manipulations, we have (see [7, Sec. IV])

$$\tilde{e}_h = \sum_{r=1}^{h} r \sum_{s=0}^{r-1} (-1)^s 2^{\binom{s}{2}} \begin{bmatrix} k \\ r \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} \binom{2^{r-s}-1}{h}. \tag{18}$$

Example 1 (Simplex Code with $k = 3$) *The $[7, 3]$ simplex code has*

$$\begin{aligned} [\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_7] &= [7, 42, 98, 105, 63, 21, 3] \\ [e_1, e_2, \ldots, e_7] &= [1, 2, 2.8, 3, 3, 3, 3]. \end{aligned} \tag{19}$$

Suppose that $n = 0$, so inserting (19) into (15) gives

$$I_E(p) = 1 - \frac{1}{7}\left[7\,p^6 + 42\,\overline{p}\,p^5 + 84\,\overline{p}^2 p^4 + 28\,\overline{p}^3 p^3\right] \tag{20}$$

where $\overline{p} = 1 - p$. This curve is plotted in Fig. 2.

Example 2 (Uncoded Transmission) *Consider the uncoded transmission of k bits. We use [9, eq. (7)] to compute*

$$A_i^r = \binom{k}{i} \sum_{s=0}^{i} (-1)^s \begin{bmatrix} i-s \\ r \end{bmatrix} \binom{i}{s}. \tag{21}$$

3.3. Split Support Weights

The fact that $\tilde{e}_h$ can be expressed in terms of the A_i^r motivates the question whether $\tilde{e}_{g,h}$ can be written in terms of appropriate generalizations of the A_i^r. This is indeed possible, as we proceed to show.

Consider again the linear code $\mathcal{C}$ formed by all pairs $(\underline{v}, \underline{x})$ in Fig. 1. We define the *split support weights* $A_{i,j}^r$ of $\mathcal{C}$ as the number of unique subspaces of $\mathcal{C}$ that have dimension r, support weight i in the first m positions of $\mathcal{C}$, and support weight j in the last n positions of $\mathcal{C}$. We prove the following generalization of (16).

Theorem 2

$$\tilde{e}_{g,h} = \sum_{r=1}^{g+h} r \sum_{s=0}^{r} (-1)^s \, 2^{\binom{s}{2}} \begin{bmatrix} k-r+s \\ s \end{bmatrix} \sum_{i=0}^{m} \sum_{j=0}^{n} \binom{m-i}{g} \binom{n-j}{h} A_{i,j}^{k-r+s} \tag{22}$$

Example 3 (Identity and Simplex Code) *Suppose Encoder 1 transmits the k information bits without coding, and Encoder 2 generates the $[2^k - 1, k]$ simplex code. We have $m = 2^k - 1$ and $n = k$, and compute*

$$A_{i,j}^r = \begin{cases} A_j^r \text{ of (21)} & \text{if } i = 2^k - 2^{k-r} \\ 0 & \text{else.} \end{cases} \tag{23}$$

For example, for $k = 3$ we use (23) in (22) to obtain the split information functions. We then use the split information functions in (15) to compute the EXIT curve to be

$$\begin{aligned} I_E(p,q) = 1 - \frac{1}{7}\{ & \\ q^3[& \; 7\,p^6 + \; 42\,\overline{p}\,p^5 + \; 84\,\overline{p}^2 p^4 + \; 28\,\overline{p}^3 p^3 \;]+ \\ \overline{q}\,q^2[& \; 18\,p^6 + \; 90\,\overline{p}\,p^5 + 108\,\overline{p}^2 p^4 + \; 36\,\overline{p}^3 p^3 \;]+ \\ \overline{q}^2 q[& \; 12\,p^6 + \; 36\,\overline{p}\,p^5 + \; 36\,\overline{p}^2 p^4 + \; 12\,\overline{p}^3 p^3 \;]\} \end{aligned} \tag{24}$$

where $\overline{p} = 1 - p$ and $\overline{q} = 1 - q$. This curve is plotted for various q in Fig. 3 as the solid lines. We recover (20) by using $q = 1$.

3.4. Duality Property

Consider the linear code $\mathcal{C}$ formed by all pairs $(\underline{v}, \underline{x})$ in Fig. 1. Let $I_E^{\perp}(\cdot)$ be the EXIT function of the dual code $\mathcal{C}^{\perp}$ of $\mathcal{C}$, i.e., $\mathcal{C}^{\perp}$ is a $[m+n, m+n-k]$ code. We have the following result.

Theorem 3 *If the extrinsic and communication channels are BECs with respective erasure probabilities p and q, then*

$$I_E^{\perp}(p,q) = 1 - I_E(1-p, 1-q). \tag{25}$$

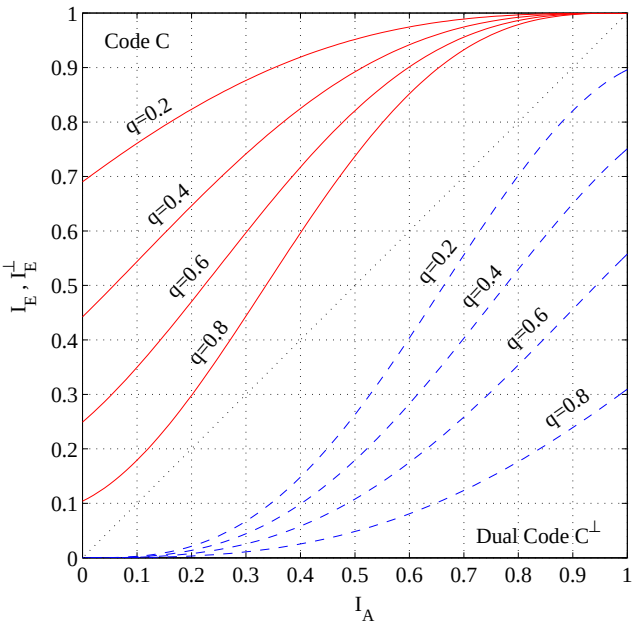

Figure 3: EXIT functions for Example 3 (solid lines) and Example 5 (dashed lines).

Observe that if there is no communication channel ($n = 0$), then we have

$$I_E^{\perp}(p) = 1 - I_E(1-p). \tag{26}$$

Example 4 (Hamming Codes) *The dual of a simplex code is a Hamming code. Consider the* $[7, 4]$ *Hamming code for which (20) and (25) give*

$$I_E(p) = \frac{1}{7}\left[7\overline{p}^6 + 42\,p\overline{p}^5 + 84\,p^2\overline{p}^4 + 28\,p^3\overline{p}^3\right]. \tag{27}$$

The curve (27) is depicted in Fig. 2. The area under this curve is $1 - R = 3/7$*, as follows from [6].*

Example 5 *Consider Example 3 and the code* $\mathcal{C}$ *formed by all pairs* $(\underline{v}, \underline{x})$*. A generator matrix for* $\mathcal{C}$ *is*

$$G = \left[\; I_k \;\middle|\; P \;\middle\|\; I_k \;\right] \tag{28}$$

where I_k *is the* $k \times k$ *identity matrix, and* P *is the* $k \times (2^k - k - 1)$ *parity check matrix of the simplex code. A generator matrix for* $\mathcal{C}^{\perp}$ *is*

$$H = \left[\begin{array}{c|c||c} P^T & I_{2^k-k-1} & 0_{(2^k-k-1)\times k} \\ I_k & 0_{k\times(2^k-k-1)} & I_k \end{array}\right] \tag{29}$$

where P^T *is the transpose of* P*, and* $0_{k\times\ell}$ *is the* $k \times \ell$ *all-zeros matrix. Thus, the situation for the dual code* $\mathcal{C}^{\perp}$ *is that Encoder 1 transmits* k *out of the* $n = 2^k - 1$ *information bits, while the generator matrix of Encoder 2 is the first of* $2^k - 1$ *columns of (29). The corresponding EXIT function can be computed using Example 3 and (25). For instance, for* $k = 3$ *we use (24) and (25) to plot the EXIT functions shown in Fig. 3 as dashed lines.*

4. Summary

Various properties of EXIT functions were derived when transmitting information over a BEC. One property expresses EXIT functions in terms of split information functions, which in turn can be computed from split weight enumerators. A duality property was derived and used to compute the EXIT function of a linear code from the EXIT function of its dual. It seems natural to suspect that other interesting EXIT properties can yet be found.

REFERENCES

[1] R. G. Gallager, *Low–density parity–check codes.* M.I.T. Press, 1963.

[2] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical Loss-Resilient Codes," in *Proc. 29th Annu. ACM Symp. Theory of Computing,* 1997, pp. 150-159.

[3] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[4] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Conf. Turbo Codes*, Brest, France, Sept. 2000.

[5] S. ten Brink, "Convergence of iterative decoding," *Electron. Lett.*, vol. 35, no. 10, pp. 806–808, May 1999.

[6] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: A model and two properties," *2002 Ann. Conf. on Inform. Sci. and Syst.*, Princeton, March 20–22, 2002.

[7] T. Helleseth, T. Kløve, and V. I. Levenshtein, "On the information function of an error-correcting code," *IEEE Trans. Inform. Theory*, vol. 43, no. 2, pp. 549–557, March 1997.

[8] T. Kløve, "Support weight distribution for linear codes," *Discrete Mathematics*, 106/107, pp. 311–316, 1992.

[9] J. Simonis, "The effective length of subcodes," *Applicable Algebra in Eng., Commun., and Computing*, vol. 5, pp. 371–377, 1994.