Computing the threshold shift for general channels

Jeremie Ezri and Rüdiger Urbanke
LTHC
EPFL
jeremie.ezri@epfl.ch, ruediger.urbanke@epfl.ch

Andrea Montanari and Sewoong Oh
Electrical Engineering Department
Stanford University
montanari@stanford.edu, swoh@stanford.edu

***Abstract*— The 'threshold' of a code ensemble can be defined as the noise level at which the block error probability curve crosses $1/2$. For ensembles of low-density parity check codes used over the binary erasure channel, the behavior of the threshold for large blocklengths is known in detail. It is characterized by an asymptotic threshold value, and a finite-blocklength shift parameter. Here we present a new method for computing the shift parameter that can be applied to general binary memoryless symmetric channels, and general message passing algorithms. We check that the new approach recovers the known parameters for erasure correction.**

I. Introduction

The *threshold* is a key notion in the asymptotic analysis of iterative coding systems. The present paper treats the definition and characterization of thresholds in finite blocklength. As shown in [1], *finite-length scaling* theory provides the correct generalization of the notion of threshold to this context and answers basic questions such as: How does the threshold change at moderate blocklength? Finite-length scaling was initially applied to erasure correction. Here we develop a computation method that allows to answer the above question on general binary memoryless symmetric (BMS) channels.

To be definite, consider communication over a smooth channel family BMS(ϵ), ordered by physical degradation with respect to the parameter $\epsilon \in [0,1]$. Canonical examples are the binary erasure channel BEC(ϵ) and the binary symmetric channel BSC(ϵ). Given an (ensemble) of codes $\mathfrak{C}(n)$ with blocklength n and a decoding algorithm, we denote by $\overline{\mathrm{P_B}}(n,\epsilon)$ its average block error probability. We define the *threshold* of such an ensemble as the smallest noise level such that the average error probability is at least one half. In formulae:

$$\epsilon_*(n) \equiv \inf\left\{\, \epsilon \quad \text{s.t.} \quad \overline{\mathrm{P_B}}(n,\epsilon) \geq 1/2 \,\right\}. \tag{1}$$

How does $\epsilon_*(n)$ behave for moderately large blocklengths (say $n = 10^2$ to 10^5)? Does it increase with n? How quickly?

Let us focus on the irregular low-density parity check code ensemble LDPC(n,λ,ρ), and assume it to be 'unconditionally stable' (i.e. $\lambda'(0)\rho'(1) < 1$.) It was conjectured in Ref. [1]

$$\epsilon_*(n) = \epsilon_* + \beta\, n^{-2/3} + o(n^{-2/3})\,. \tag{2}$$

The same paper described a method to compute the shift parameter β, for regular ensembles over the erasure channel. The method was generalized to irregular ensembles in [2]. The conjecture was subsequently proved in [3] for the case of Poisson left degrees using strong approximation techniques.

The expression (2) provides a rather sharp answer to the above questions, yielding accurate predictions already for $n \approx 10^3$. As an example, we compare the threshold values as computed from numerical simulations, with the prediction provided by the two leading terms in Eq. (2), for the $(3,6)$ regular ensemble over the BEC, under iterative decoding.

n	$\epsilon_*(n)$	$\epsilon_*^{\mathrm{asym}}(n)$
256	0.415 ± 0.001	0.414138
512	0.420 ± 0.001	0.419801
1024	0.423 ± 0.001	0.423368
2048	0.425 ± 0.001	0.425615

The motivation for developing such a sharp analytic control is to use it in ensemble optimization [2].

The infinite block-length threshold ϵ_* can be computed for general BMS channels, and message-passing decoders, using density evolution [4]. Unhappily, both the method for computing β in [1], [2], and the proof in [3] were based on the representation of the decoding process, through a finite-dimensional Markov chain [5]. Such a representation only exists for erasure decoding.

This paper develops a new method for computing β, that does not rely on the Markov chain representation, and is thus applicable to general channels. The method is described in Section II, and is based on a simplified model for the performance curve that allows to connect the parameter β to a different quantity that is more directly computable. The model is expected to be exact for large blocklengths and close to threshold. Section II also describes a way to carry out the calculation using a branching process representation.

Since the method is based on a few unproven assumptions, it is important to validate it by checking its results. In Section III we recover the known results for the BEC. Applications to other channel models will be reported in a forthcoming publication.

II. A new method

Our approach is based on a model for the 'performance curves.' Consider, to be definite, a quantized message passing decoder, where the 'soft decision' for i-th bit, to be denoted by ν_i, takes values in a finite alphabet $\mathcal{M}$, and assume that the all-zeros codeword has been transmitted. Given a function $f : \mathcal{M} \to \mathbb{R}$, and the set of decisions $\{\nu_i^{(t)}\}$ achieved after t message passing iterations, we define the performance value

$$h_n(\epsilon) = \frac{1}{n}\sum_{i=1}^{n} f(\nu_i^{(t)})\,. \tag{3}$$

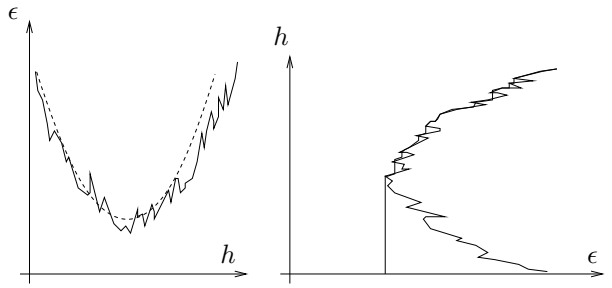

Fig. 1. Sketch of the performance curve for a given code and channel realization. Left: channel parameter ϵ as a function of the performance parameter. The dashed line corresponds to the deterministic component $h_* + \frac{1}{2}A(\epsilon - \epsilon_*)^2 + \dots$. Right: performance versus noise. The actual performances follow the upper branch (thick line.)

Well-known examples would be the EXIT or GEXIT functions, or the bit-error rate. Throughout, we will assume t to be a very large value, such that message passing process has reached a stationary state (not necessarily a fixed point [6].)

For any $\epsilon \in [0,1]$, the quantity $h_n(\epsilon)$ is a random variable because of the graph and of the channel randomness. We want instead to think the whole function $h_n : [0,1] \to \mathbb{R}$ as a random process, i.e. to define $h_n(\epsilon)$ and $h_n(\epsilon')$ on the same probability space for any two ϵ, ϵ' (to *couple* them.)

There is a 'natural' way for doing this. First draw a random factor graph from the code ensemble: this will be used for all the values of ϵ. Then recall that the channel family BMS(ϵ) is ordered by physical degradation. For any two values of the noise levels $\epsilon < \epsilon + \delta$ there exists a channel C(ϵ, δ) such that transmitting through BMS($\epsilon + \delta$) is equivalent to transmitting through BMS(ϵ) and then C(ϵ, δ). Therefore, one can couple channel realizations on an arbitrarily fine mesh $\epsilon_1, \epsilon_2, \dots, \epsilon_K$ by transmitting in sequence through BMS(ϵ_1), C($\epsilon_1, \epsilon_2 - \epsilon_1$), C($\epsilon_2, \epsilon_3 - \epsilon_2$)$\dots$. This defines $h_n(\epsilon)$ on the mesh. The full function can be defined through a limiting procedure (a more formal definition uses Markov semigroups [7].)

Let us finally mention that the curve $h_n(\epsilon)$ can also be defined in terms of messages, to be denoted by $\nu^{(t)}_{i\to a}$ (variable to check node) or $\widehat{\nu}^{(t)}_{a\to i}$, for instance as (here the sum runs over all the edges in the factor graph) $h_n(\epsilon) \equiv n^{-1}\sum_{(i,a)\in E} f(\nu^{(t)}_{i\to a})$. We shall further discuss this point in Section III.

A. Shift from variance

In the infinite block-length limit, the performance curve is expected to converge to a non-random limit $h(\epsilon)$. For unconditionally stable ensembles, this limit curve exhibit a square root singularity at the critical point: $h(\epsilon) = h_* + \Theta(\sqrt{\epsilon - \epsilon_*})$. It is therefore natural to think of ϵ as a function of h with (in the simplest case) a decreasing and an increasing branch: the inverse of the latter give rise to the function $h(\epsilon)$. Near the critical point we have $\epsilon(h) = \epsilon_* + \frac{1}{2}A(h-h_*)^2 + O((h-h_*)^3)$. Our model for the performance curve at blocklength n is (up to higher order terms)

$$\epsilon_n(h) \simeq \epsilon_* + \frac{A}{2}\,(h-h_*)^2 + \frac{\alpha}{\sqrt{n}}\,X + \frac{\gamma}{\sqrt{n}}\,W(h-h_*)\,. \quad (4)$$

Here X is a standard normal random variable, and $W(\,\cdot\,)$ a two-sided brownian motion. The performance curve is obtained by selecting the upper branch of this curve: $h_n(\epsilon) = \sup\{h : \epsilon_n(h) \le \epsilon\}$. This behavior is sketched in Fig. 1.

Assuming this model, we can expand the error probability $\overline{\mathrm{P_B}}(n,\epsilon) \equiv \mathbb{P}\{\min_h \epsilon_n(h) \le \epsilon\}$ for $\epsilon = \epsilon_* + zn^{-1/2}$. Using the results of Ref. [8] on the distribution of the minimum of a Brownian motion with parabolic drift, we get

$$\overline{\mathrm{P_B}}(n,\epsilon) = Q(z/\alpha) + \frac{\gamma^{4/3}\Omega}{A^{1/3}\alpha n^{1/6}}\,Q'(z/\alpha) + O(n^{-1/3})\,.$$

where Ω is an universal constant[1] already introduced in [1]. We improved our numerical evaluation of this constant obtaining $\Omega = 0.996 \pm 0.001$.

Matching the above expression for $\overline{\mathrm{P_B}}(n,\epsilon)$ with the definition of the shift parameter, cf. Eq. (2), we get

$$\beta/\Omega = \gamma^{4/3}A^{-1/3}\,. \quad (5)$$

The asymptotic curve $h(\epsilon)$ and hence the constant A can be evaluated using density evolution. In order to estimate γ, consider two noise levels ϵ and $\epsilon + \Delta\epsilon$, and let $\Delta h \equiv h_n(\epsilon + \Delta\epsilon) - h_n(\epsilon)$. It follows from our model that $\Delta h \simeq h'(\epsilon)\,\Delta\epsilon + \gamma h'(\epsilon)^{3/2} Z\,\sqrt{\Delta\epsilon/n}$, where Z is a standard normal random variable. Therefore γ is determined as:

$$\gamma^2 = \lim_{\epsilon\downarrow\epsilon_*}\lim_{\Delta\epsilon\to 0}\lim_{n\to\infty} (2A)^{3/2}(\epsilon-\epsilon_*)^{3/2}\,\frac{n\mathrm{Var}(\Delta h)}{\Delta\epsilon}\,.$$

In this formula Δh can be interpreted as the change in $h_n(\epsilon)$ when each symbol y_i of the received message is transmitted through the channel C($\epsilon, \Delta\epsilon$) defined above. In order to obtain a more explicit expression, let ΔH_i denote the change in $nh_n(\epsilon)$ when the symbol received at position i is transmitted through C($\epsilon, \Delta\epsilon$). Then we expect

$$\mathrm{Var}(\Delta h) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}(\Delta H_i) + o(\Delta\epsilon)\,.$$

If the channel family is smooth, we can also take the limit $\Delta\epsilon \to 0$. Let $R_{\epsilon,\Delta\epsilon}(z|y)$ be the transition probability kernel for $C(\epsilon, \Delta\epsilon)$ (for notational simplicity, we assume the channel to be discrete.) For small $\Delta\epsilon$, this admits the Taylor expansion $R_{\epsilon,\Delta\epsilon}(z|y) = \mathbb{I}(z = y) + \Delta\epsilon\, L_\epsilon(z|y) + o(\Delta\epsilon)$. If $\Delta H_i(z)$ denotes the change in $nh_n(\epsilon)$ when the received symbol at i is changed to z, we get, for $b = 1, 2$

$$\mathbb{E}\{(\Delta H_i)^b\} = \Delta\epsilon\sum_z \mathbb{E}\{L_\epsilon(z|y_i)\Delta H_i(z)^b\} + o(\Delta\epsilon)\,.$$

Using these results in the expression for γ^2 we get

$$\gamma^2 = \lim_{\epsilon\downarrow\epsilon_*}(2A)^{3/2}(\epsilon-\epsilon_*)^{3/2}\sum_z \mathbb{E}\{L_\epsilon(z|y_i)\Delta H_i(z)^2\}\,. \quad (6)$$

Here it is understood that the expectation of $\Delta H_i(z)^2$ is computed in the large blocklength limit. If changing the

[1] Defined by $\Omega = \int_0^\infty [1-\mathcal{K}(z)]^2 \mathrm{d}z$ and

$$\mathcal{K}(z) \equiv \int \frac{\mathrm{Ai}(iu)\mathrm{Bi}(2^{1/3}z+iu) - \mathrm{Ai}(2^{1/3}z+iu)\mathrm{Bi}(iu)}{2\mathrm{Ai}(iu)}\,\mathrm{d}u\,.$$

received value at node i implies a bounded number of changes in other decisions, this expectation only involves the local structure of the factor graph that defines the code. As a consequence, its limit is obtained through a tree computation.

In Section II-C we'll see how to do this computation, and hence obtain β via a branching process representation. Before this, a brief intermezzo on this topic is probably helpful.

B. Multi-type branching processes

A d-type branching process is a Markov chain with state space $\mathbb{N}^d$. It is defined by d random variables $X^{(1)}$, $X^{(2)}$, ..., $X^{(d)}$ with values in $\mathbb{N}^d$ as well, Given the current state $Z_t = (Z_{t,1}, \dots, Z_{t,d})$, its state at time $t+1$ is

$$Z_{t+1} = \sum_{i=1}^{d} \sum_{j=1}^{Z_{t,i}} X^{(i)}(j)\,, \tag{7}$$

where the $X^{(i)}(j)$'s are iid copies of $X^{(i)}$. The initial condition will be $Z^{(0)} \stackrel{\mathrm{d}}{=} W$, for some random variable W.

The trajectory of a branching process can be represented by a rooted tree whose vertices take d distinct colors. The defining property of a branching process is the following. The number of offsprings of each vertex, and their colors is conditionally independent from the rest of the tree, given the parent's color.

We will be interested in branching processes that terminate almost surely. The total number of elements of type i in the process, defined as $T_i = \sum_{t\geq 0} Z_{t,i}$, is then almost surely finite. A special role is played by the $d \times d$ matrix K with entries $K_{ij} \equiv \mathbb{I}_{i,j} - M_{ij}$ where $M_{ij} \equiv \mathbb{E}\{X_j^{(i)}\}$. The expectation of $T = (T_1, T_2, \dots, T_d)$ (that is a vector in $\mathbb{R}_+^d$) and its covariance (a $d \times d$ matrix) are then given by

$$\begin{aligned} \mathbb{E}\{T\} &= (K^T)^{-1}\,\mathbb{E}\{W\}\,, \\ \mathrm{Cov}(T,T) &= \mathrm{Cov}(\tilde{W},\tilde{W}) + \sum_{i=1}^{d} \mathbb{E}\{T_i\}\mathrm{Cov}(\tilde{X}^{(i)},\tilde{X}^{(i)})\,. \end{aligned}$$

where $\tilde{W} \equiv (K^T)^{-1}W$, $\tilde{X}^{(i)} \equiv (K^T)^{-1}X^{(i)}$.

The branching process terminates almost surely if $||\rho(M)|| < 1$. We will consider the case in which the largest of its singular values is close to 1. Call ξ this value, let $u, v \in \mathbb{R}^d$ be its left and right eigenvectors (with $u^T v = 1$), and $T_* \equiv g^T T$ for some vector $g \in \mathbb{R}^d$. Then $\mathbb{E}\{T_*\} \simeq C_1(1-\xi)^{-1}$, $\mathrm{Var}(T_*) \simeq C_2(1-\xi)^{-3}$ (here $\simeq$ indicates equality up to lower order terms as $\xi \to 1$) and

$$C_1 = (u^T g)\mathbb{E}\{v^T W\}\,, \tag{8}$$

$$C_2 = (u^T g)^2 \mathbb{E}\{v^T W\} \sum_{i=1}^{d} u_i \mathrm{Var}(v^T X^{(i)})\,. \tag{9}$$

C. Variance computation through branching processes

Consider the computation of $\mathbb{E}\{\Delta H_i(z)^2\}$ that enters the definition of γ^2, cf. Eq. (6). With the notation of the previous Section, we will show that $\Delta H_i(z) = T_*$ for a properly chosen branching process and vector g. The branching process will be identified by describing its representative tree, and showing that offsprings at any two different vertices are independent.

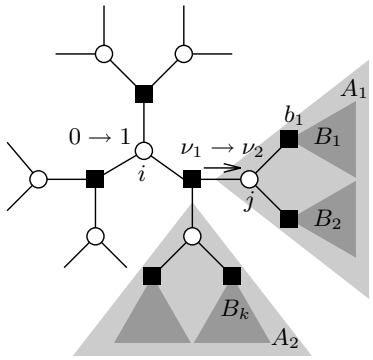

Fig. 2. The neighborhood of variable node i in the factor graph. Changing the received symbol from 0 to 1 induces a change in one of the outgoing messages from ν_1 to ν_2. In this case $W = 1$. In Section III-B, to compute the variance recursively, the factor graph is divided into subtree B_i's.

First, as stressed above the computation can be done by assuming the factor graph is an infinite random tree, rooted at i, whose distribution models the local structure of the original graph. Consider, to be specific, random codes from the LDPC(λ, ρ, n) ensemble (here λ, ρ are the variable and check nodes degree distributions from edge perspective.) Then the tree is itself a branching process with offspring distributions λ (for variable nodes) and ρ (check nodes) except at the root whose degree distribution (generating polynomial) is $\Lambda(x) = \int_0^x \lambda(u)\mathrm{d}u / \int_0^1 \lambda(u)\mathrm{d}u$.

Consider a realization of the channel output $\underline{y} = \{y_j\}$ (whereby each variable node is associated a symbol y_j), and the one obtained by replacing y_i with z. Imagine to run in parallel the message passing decoder for these two realizations. At time t, a subset of the outbound messages will differ between the two, and a subset of the decisions on nodes adjacent to those messages will differ as well. Focus, to be definite, on check-to-variable messages. The branching process we want to construct has vertices in correspondence with the messages of this type that differ in the two realizations. The 'color' of each vertex encodes the type of change. More precisely, assume that both messages and decisions take values in the same finite alphabet $\mathcal{M}$, and let $a \to j$ be a check-to-variable edge whose message change. If $\hat{\nu}_{a\to j}^{(t)} = \nu_1$ in the first realization and ν_2 in the second, while the decision is $\nu_j^{(t)} = \bar{\nu}_1$ in the first realization and $\bar{\nu}_2$ in the second, then the corresponding vertex in the branching process will be of type $(\nu_1, \nu_2, \bar{\nu}_1, \bar{\nu}_2)$. Therefore the number of types is $d = |\mathcal{M}|^2(|\mathcal{M}|-1)^2$. This can be significantly reduced in specific cases.

It is convenient to denote by $T_*(y,z)$ a random variable distributed as $\Delta H_i(z)$ conditional on $y_i = y$. Its moments can be computed along the lines discussed in the previous Section provided:

(1) The initial condition $W(y,z)$ is distributed as the number of changes in the messages outgoing from the check nodes adjacent to i, cf. Fig. 2.

(2) The random variable $X^{(j)}$ is distributed as the number of changes in messages outgoing from check nodes at distance $\ell+1$ from i, when their parent was of type $j = (\nu_1, \nu_2, \bar{\nu}_1, \bar{\nu}_2)$

(3) The vector g has entries $g(\nu_1,\nu_2,\bar{\nu}_1,\bar{\nu}_2) = f(\nu_2) - f(\nu_1)$ (if the performance curve is defined with respect to messages) or $= f(\bar{\nu}_2) - f(\bar{\nu}_1)$ (if it is defined with respect to nodes.)

We shall denote by $C_{1,2}(y,z)$ the constants defined in Eqs. (8), (9). We then have $\frac{\mathrm{d}h}{\mathrm{d}\epsilon} = \sum_z \mathbb{E}\{L_\epsilon(z|y_i)T_*(y_i,z)\} \simeq \sum_z \mathbb{E}\{L_\epsilon(z|y_i)C_1(y_i,z)\}(1-\xi)^{-1}$. This expression provides a relation between $C_1 \equiv \sum_z \mathbb{E}\{L_\epsilon(z,y_i)C_1(y_i,z)\}$ and the constant A introduced in Section II-A. Using this relation in Eqs. (5) and (6) in conjunction with the expressions (8) and (9) we obtain our final formula

$$(\beta/\Omega)^{3/2} = \frac{\sqrt{2}S}{\mathbb{E}\{v^T\overline{W}\}} \sum_{i=1}^{d} u_i \mathrm{Var}(v^T X^{(i)})\,, \qquad (10)$$

where $\overline{W} \equiv \sum_z L_\epsilon(z|y_i)W(y_i,z)$ and $S \equiv \lim_{\epsilon\downarrow\epsilon_*}\sqrt{\epsilon-\epsilon_*}/(1-\xi(\epsilon))$.

III. The binary erasure channel revisited

In this Section we apply the formulae derived above to the BEC(ϵ), and check that the result agrees with the ones of [1], [2]. We further recompute the variance of the number of changes in Section III-B through an elementary method that does not use the branching process representation. In both approaches it turns out to be important to define the performance curve $h_n(\epsilon)$ through node rather than message quantities. We suspect this to be a peculiarity of the erasure channel, related to the fact that a message can change value without any influence on the node reliabilities. We shall further discuss this point in [7].

For reference the asymptotic threshold and shift parameters for a few (l,k) regular ensembles are: $(3,4)$, $\epsilon_* \approx 0.6473$, $\beta/\Omega \approx 0.5936$; $(3,5)$, $\epsilon_* \approx 0.5176$, $\beta/\Omega \approx 0.6161$; $(3,6)$, $\epsilon_* \approx 0.4294$, $\beta/\Omega \approx 0.6169$; $(4,5)$, $\epsilon_* \approx 0.6001$, $\beta/\Omega \approx 0.5716$; $(6,6)$, $\epsilon_* \approx 0.5061$, $\beta/\Omega \approx 0.5743$; $(5,6)$, $\epsilon_* \approx 0.5510$, $\beta/\Omega \approx 0.5596$.

The general formula for regular ensembles is

$$\frac{\beta}{\Omega} = \left(\frac{l-2}{lb_*^l}\right)^{2/3} \epsilon_*^{1/3}\left[\frac{l}{l-1} + \frac{(k-2)a_*}{1-a_*} - 2\right]^{-1/3}\,, \qquad (11)$$

where $a_*, b_* > 0$ solve the density evolution equations $a = \epsilon b^{l-1}$, $b = 1-(1-a)^{k-1}$ at the critical point ϵ_*.

A. Shift via branching processes

Two simplifications arise for the BEC: (i) Messages only take two values $\{*,0\}$ and only change from $*$ to 0 if noise is decreased. (ii) The branching process keeps its conditional independence property even if the type only involves the node decision. As a consequence, the branching process has $d = 1$ type of vertices, corresponding to variable nodes whose decision changes from $*$ to 0.

Formally, we have a single scalar variable X, and a scalar W. Two channel realizations for BEC(ϵ) and BEC($\epsilon+\Delta\epsilon$) are coupled using an erasure channel with kernel $R_{\epsilon,\Delta\epsilon}(*|*) = 1$ and $R_{\epsilon,\Delta\epsilon}(s|s) = 1-\Delta\epsilon/(1-\epsilon)$, $R_{\epsilon,\Delta\epsilon}(*|s) = \Delta\epsilon/(1-\epsilon)$, whence $\mathbb{E}\{v^T W\} = \mathbb{E}\,W(0,*)$.

In order to compute $\mathbb{E}W(0,*)$, notice that i has, on average, $\Lambda'(1)\rho'(1)$ neighboring variable nodes. Consider one of them, call it j, and call c the function node between i and j. Denote by a (resp. b) the probability that variable-to-check (check-to-variable) messages are erased, at the density evolution fixed point. Recall that these quantities satisfy the equations $a = \epsilon\lambda(b)$, $b = 1-\rho(1-a)$.

The probability that the decision at j changes when the received value at x changes is the product of three factors: (i) the probability that the message from i to c is $*$ when the received message at i is $*$ (equal to a); (ii) the probability that all the messages coming to c from variable nodes different from i and j are 0's (this is $\rho'(1-a)$); (iii) The probability that the received symbol at j is $*$ alongside with all the messages to j different from the one from c (this is $\lambda(b)$). Collecting these factors, we get:

$$\mathbb{E}\,W(0,*) = \Lambda'(1)\lambda(b)a\rho'(1-a)\,. \qquad (12)$$

Arguing along the same lines, one can show that $X \stackrel{\mathrm{d}}{=} \sum_{i=1}^{M} B_i$ where the B_i's are iid Bernoulli random variables with mean $a\rho'(1-a)/[1-\rho(1-a)]$, and $\mathbb{P}\{M=m\} = \lambda_{m+1}b^m/\lambda(b)$. It is then straightforward to compute the first two moments:

$$\mathbb{E}\{X\} = \epsilon\lambda'(b)\rho'(1-a)\,, \qquad (13)$$

$$\mathrm{Var}(X) = \epsilon\lambda'(b)\rho'(1-a)\,(1-a\rho'(1-a)/b) + \qquad (14)$$
$$+(a\rho'(1-a)/b)^2\left(\frac{b^2\lambda''(b)}{\lambda(b)} + \frac{y\lambda'(b)}{\lambda(b)} - \frac{b^2\lambda'(b)^2}{\lambda(b)^2}\right)\,.$$

Obviously $\xi(\epsilon) = \epsilon\lambda'(b)\rho'(1-a)$, whence by simple calculus

$$S = \sqrt{2a_*[\lambda''(a_*)\rho'(1-a_*)^2 - \lambda(b_*)\rho''(1-a_*)]}\,. \qquad (15)$$

(here a_*, b_* refer to the density evolution fixed point at $\epsilon = \epsilon_*$.) Finally, the shift parameter is obtained by specializing Eq. (10) to the present case $\beta/\Omega = \{\sqrt{2}S\,\mathrm{Var}(X)/\mathbb{E}W(0,*)\}^{2/3}$, and substituting Eqs. (12), (14) and (15). The result can be shown to coincide with the one reported in [2] and with Eq. (11) for regular ensembles.

B. Shift via correlations

Consider a regular (l,k) ensemble and let ΔH_i denote the number of variable nodes whose decisions change when the received symbol at root node i passes from $*$ to 0. In order to check the branching process representation, and to develop an alternative route, we computed $\mathrm{Var}(\Delta H_i)$ via a more direct recursive approach. The result can be used to compute β through Eqs. (6) and (5). In Fig. 3 we compare the result of this calculation with numerical simulations on the $(3,6)$ ensemble.

Let us now sketch the direct approach, focusing on the leading terms as $\epsilon \downarrow \epsilon_*$. Denote by $\hat{x}_j(\underline{y})$ the message passing decision at variable node j when the vector of received symbols is $\underline{y}$. Further, let $\underline{y}^{(0)}$, $\underline{y}^{(*)}$ be the same vector, where y_i has been replaced (respectively) by 0 and $*$. To each variable node j we associate the value

$$\mathsf{f}(j) \equiv \mathbb{I}\{\hat{x}_j(\underline{y}^{(0)}) = 0\} - \mathbb{I}\{\hat{x}_j(\underline{y}^{(*)}) = 0\}\,. \qquad (16)$$

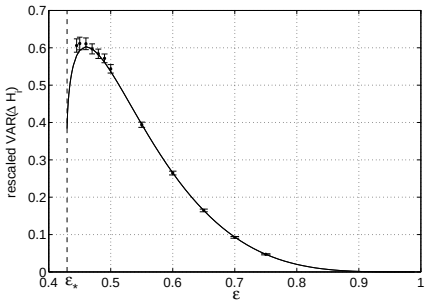

Fig. 3. Rescaled variance, $\mathrm{Var}(\Delta H_i)(1-\xi(\epsilon))^3$, for the $(3,6)$ regular ensemble over BEC(ϵ). Comparison of the variance determined via simulations (dots with 95% confidence intervals) with the prediction given by the recursive approach. The blocklength is $n = 50000$.

For any set of variable nodes $U \subseteq V$, we let $\mathsf{f}(U) = \sum_{j\in U} \mathsf{f}(j)$. Then we have $\Delta H_i = \mathsf{f}(V)$.

As above, we shall compute $\mathrm{Var}(\mathsf{f}(V))$ on a tree rooted at node i, cf. Fig. 2. The root node has l neighboring check nodes denoted by a_j, $j = 1,\dots,l$, and each of these node is the root of an identical subtree denoted by A_j. Decomposing V into these identical subtrees, we get,

$$\mathrm{Var}(\mathsf{f}(V)) = l\,\mathrm{Var}(\mathsf{f}(A_1)) + l(l-1)\,\mathrm{Cov}(\mathsf{f}(A_1),\mathsf{f}(A_2))\,. \quad (17)$$

Using the computation methods from [2], one can show that $\mathrm{Cov}(A_1,A_2) = O((1-\xi(\epsilon))^{-2})$, where $\xi(\epsilon) = \epsilon\lambda'(b)\rho'(1-a)$. This is a lower order term, hence negligible as $\epsilon \downarrow \epsilon_*$. Further, since $\mathbb{E}\{\mathsf{f}(A_1)\}$ can be shown to be of order $(1-\xi(\epsilon))^{-1}$, we can focus on computing $\mathbb{E}\{\mathsf{f}(A_1)^2\}$.

This quantity can be decomposed into further subtrees. The root of A_1 (which we called a_1) has $(l-1)(k-1)$ neighboring check nodes A_1, which we denote b_j, $j = 1,\dots,(l-1)(k-1)$, each of them being the root of subtree B_j. We thus get

$$\begin{aligned}\mathbb{E}\{\mathsf{f}(A_1)^2\} \;\simeq\; & (l-1)(k-1)\mathbb{E}\{\mathsf{f}(B_1)^2\} + \quad (18)\\ & +(l-1)(l-2)(k-1)\mathbb{E}\{\mathsf{f}(B_1)\mathsf{f}(B_2)\}\\ & +(l-1)^2(k-1)(k-2)\mathbb{E}\{\mathsf{f}(B_1)\mathsf{f}(B_k)\}\,.\end{aligned}$$

Here it is understood that the second term in the expansion involves correlations between subtrees whose roots are offsprings of the same variable node, and third term involves correlations between subtrees whose roots are offsprings of two distinct variable nodes. Further we neglected the contribution of the variable node that is in A_1 but not in B_j as it is again of lower order.

Since only one of the messages coming out of a_1 can change as the message received at the root node i changes from 0 to $*$, $\mathbb{E}\{\mathsf{f}(B_1)\mathsf{f}(B_k)\}$ vanishes.

To compute the first term, observe that the trees A_1 and B_1 have same structure and only differ in the distance from the root node i. Let the variable node between a_1 and b_1 be denoted j. Conditional on the event that the messages from i to a_1, $\nu_{i\to a_1}$, and from j to b_1, $\nu_{j\to b_1}$, change as the received message at the root node changes, $\mathsf{f}(A_1)$ and $\mathsf{f}(B_1)$ are identically distributed.

The probability that $\nu_{j\to b_1}$ changes when the received value at i changes is the product of three factors: (i) the probability all the incoming messages to i from check nodes other than a_1 are $*$'s (this is b^{l-1}); (ii) the probability that all the incoming messages to a_1 from variable nodes different from i and j are 0's (this is $(1-a)^{k-2}$); (iii) the probability the received symbol at j with all the incoming messages from check nodes other than a_1 and b_1 are $*$'s (this is ϵb^{l-2}). Collecting these terms, using the density evolution equations, we get

$$\begin{aligned}\mathbb{E}\{\mathsf{f}(B_1)^2\} &= \frac{a^2(1-b)}{\epsilon b(1-a)}\mathbb{E}\{\mathsf{f}(A_1)^2|\mathsf{f}(\nu_{i\to a_1})=1\}\\ &= \frac{a(1-b)}{b(1-a)}\mathbb{E}\{\mathsf{f}(A_1)^2\}\,, \quad (19)\end{aligned}$$

where the second identity can be shown to follow from the definition of $\mathsf{f}(\nu_{i\to a_1})$.

The second term in Eq. (18) is treated similarly Notice that $\mathsf{f}(B_1)$ and $\mathsf{f}(B_2)$ are conditionally independent given that messages from j to b_1 and b_2 change. Further $\mathbb{E}\{\mathsf{f}(B_1)|\mathsf{f}(\nu_{i\to a_1}) = 1\}$ can be computed using the methods from [2],

$$\begin{aligned}\mathbb{E}\{\mathsf{f}(B_1)\mathsf{f}(B_2)\} &= \frac{a^2(1-b)}{\epsilon(1-a)}\mathbb{E}\{\mathsf{f}(B_1)|\mathsf{f}(\nu_{i\to a_1})=1\}^2\\ &= \frac{a^2(1-b)}{\epsilon(1-a)(l-1)^2}\left(\frac{\xi(\epsilon)}{1-\xi(\epsilon)}\right)^2\,. \quad (20)\end{aligned}$$

With these results, to find $\mathbb{E}\{\mathsf{f}(A_1)^2\}$, substitute Eqs. (19) and (20) to (18), and solve the recursion. Neglecting lower order terms we get

$$\mathbb{E}\{\mathsf{f}(A_1)^2\} \simeq \frac{(k-1)(l-2)}{(l-1)}\frac{a^2(1-b)}{\epsilon(1-a)}\frac{\xi(\epsilon)^2}{(1-\xi(\epsilon))^3}\,. \quad (21)$$

Finally $\mathrm{Var}\{\Delta H_i\} = \mathrm{Var}\{\mathsf{f}(V)\} \simeq l\mathbb{E}\{\mathsf{f}(A_1)^2\}$ plus lower order terms.

Acknowledgment

A.M. is partially supported from a David Filo and Jerry Yang fellowship.

References

[1] A. Amraoui, A. Montanari, T. J. Richardson and R. Urbanke, "Finite-Length Scaling for Iteratively Decoded LDPC Ensembles," IEEE Trans. on Inf. Theory, accepted, ArXiv: cs.IT/0406050

[2] A. Amraoui, A. Montanari and R. Urbanke, "How to Find Good Finite-Length Codes: From Art Towards Science,' Eur. Trans. on Telecomms., 18 (2007) 491-508

[3] A. Dembo and A. Montanari, "Finite Size Scaling for the Core of Large Random Hypergraphs," Annals of Appl. Prob., 2007, accepted.

[4] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge University Press, 2008

[5] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," IEEE Trans. Inform. Theory, **47** (2001), 569–584.

[6] J. Ezri, A. Montanari, and R. Urbanke, "A Generalization of the Finite-Length Scaling Approach Beyond the BEC," Proc. of IEEE Int. Symp. on Inf. Theory, Nice, France, July 2007

[7] J. Ezri, A. Montanari, S. Oh, and R. Urbanke, "Finite-Length Scaling for General Memoryless Chanels," in preparation.

[8] P. Groeneboom, "Brownian motion with a parabolic drift and Airy functions," Probab. Th. Rel. Fields, **81** (1989), 79–109.