On the Construction of Polar Codes for Channels with Moderate Input Alphabet Sizes

Ido Tal, *Member, IEEE*

***Abstract*—Current deterministic algorithms for the construction of polar codes can only be argued to be practical for channels with small input alphabet sizes. In this paper, we show that any construction algorithm for channels with moderate input alphabet size which follows the paradigm of "degrading after each polarization step" will inherently be impractical with respect to a certain "hard" underlying channel. This result also sheds light on why the construction of low-density parity-check (LDPC) codes using density evolution is impractical for channels with moderate sized input alphabets.**

***Index Terms*—Polar codes, LPDC, construction, density evolution, degrading cost.**

I. Introduction

Polar codes [1] are a novel family of error correcting codes that achieve capacity and have efficient encoding and decoding algorithms. Originally defined for channels with binary input, they were soon generalized to channels with arbitrary input alphabets [2]. Although polar codes are applicable to many information-theoretic settings, the channel coding setting is the one we consider in this paper. More specifically, we consider the symmetric capacity setting discussed in [1] and [2].

The "plus" and "minus" polar transforms were defined in the seminal paper [1]. We now briefly state these, and mention that other transforms are possible: [3], [4] [5], see also [6]. Let $W : \mathcal{X} \to \mathcal{Y}$ be a channel with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$. Denote by $W(y|x)$ the probability that $y \in \mathcal{Y}$ was received given that $x \in \mathcal{X}$ was transmitted. Then, the minus transform $W^- : \mathcal{X} \to \mathcal{Y}^2$ and plus transform $W^+ : \mathcal{X} \to \mathcal{Y}^2 \times \mathcal{X}$ are defined respectively as

$$W^-(y_1, y_2|x_1) = \sum_{x_2 \in \mathcal{X}} \frac{1}{|\mathcal{X}|} W(y_1|x_1 + x_2)W(y_2|x_2) \ ,$$

$$W^+(y_1, y_2, x_1|x_2) = \frac{1}{|\mathcal{X}|} W(y_1|x_1 + x_2)W(y_2|x_2) \ .$$

The operation $x_1 + x_2$ is defined in [1] for the binary case $\mathcal{X} = \{0, 1\}$ as addition modulo 2. This is generalized in [2] by considering addition modulo $|\mathcal{X}|$, where $|\mathcal{X}|$ is a prime. In this setting, a polar code of length $n = 2^m$ is conceptually gotten as follows. We consider all the synthesized channels one can get by starting with the underlying channel, and applying m consecutive polar transforms to it, each transform being either a minus transform or a plus transform. It is easily seen that there are $n = 2^m$ such channels. The polar code is gotten by specifying a suitably chosen constant $\beta > 0$, and choosing the synthesized channels with probability of error at most 2^{-n^β}. This set of channels defines an error correcting code. As m tends to infinity, the rate of the code approaches the symmetric capacity of the underlying channel, while the probability of misdecoding is bounded by $n \cdot 2^{-n^\beta}$. That is, it approaches 0.

An important point to note about the above is that if the input alphabet of the channel W is $\mathcal{Y}$, then the output alphabet of W^- and W^+ are $\mathcal{Y}^2$ and $\mathcal{Y}^2 \times \mathcal{X}$, respectively. Namely, each polar transform at least squares the output alphabet size. Thus, the $n = 2^m$ synthesized channels have an output alphabet size which grows exponentially in the code length n. We conclude that calculating their probability of misdecoding is intractable, if approached directly. To the author's knowledge, the only tunable and deterministic methods of circumventing this difficulty involve approximating some of the intermediate channels by channels which have a manageable output alphabet size. Simply put: before the first polarization step and after each polarization step, approximate the relevant channel by another channel having a prescribed output alphabet size. Doing so ensures that the channel output alphabet sizes do not grow intractably.

The above "approximate after each polarization step" idea has its origins in density evolution [7, Page 217], a method to evaluate the performance of LDPC code ensembles. Indeed, when analyzing LDPC codes via density evolution, we are essentially applying generalized plus and minus transforms: we start with the underlying channel and apply a generalized minus transform followed by a generalized plus transform. The resulting channel is again subjected to these two consecutive transforms, and the process is repeated a specified number of times. Density evolution was suggested as a method of constructing polar codes in [8]. In order to bound the misdecoding probability of a synthesized channel one can force the approximating channel to be either (stochastically) degraded or upgraded with respect to it.

An efficient algorithm for such a degrading/upgrading approximation was introduced for the binary-input case in [9] and analyzed in [10]. The degrading algorithm in [9] is easy to explain: as long as the output alphabet size is above a required threshold, we select two output letters and merge them, in the sense of making them indistinguishable. This is applied also to their symmetric conjugates. The merged letters are selected so that the drop in mutual information between channel input and channel output is minimized, assuming a uniform input distribution. This procedure results in the output alphabet size decreasing by two, and is repeated until the required threshold is met. The upgrading algorithm in [9] is similar, in the sense that a "greedy" criterion is applied to selecting the three best

This work was supported by the Israel Science Foundation under grant 1769/13. This paper was presented in part at the 2015 IEEE International Symposium on Information Theory, Hong Kong.

The author is with the Department of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 32000 (email:idotalee.technion.ac.il).

output letters to be reduced to two letters. We mention also [11], in which an optimal degrading algorithm is introduced.

Algorithms for degrading and upgrading non-binary channels were given in [12] and [13], respectively. The general idea in these algorithms is to merge output letters that fall into the same "bin". Essentially, two output letters are in the same bin if the posterior probabilities associated with the various inputs are "close". Merging all the letters in the bin as described above leads to a degraded channel. An upgrading "merge" is more involved, but possible. We also mention [14], in which a generalization of the pairs-merging idea introduced in the preceding paragraph is applied to a non-binary setting. On a related note, the construction of polar codes was recently proven to be polynomial in the blocklength [15], for an arbitrary but *fixed* input alphabet size. The result in [16] is relevant as well.

For a fixed input distribution, a degrading approximation results in a channel with reduced mutual information between input and output. This drop in mutual information should ideally be kept small. The reason for this will be elaborated on in Section III. In brief, the reason is that such a drop necessarily translates into a drop in code rate, both in the polar coding setting as well as in the LDPC setting. Thus, a non-negligible drop in mutual information due to approximation necessarily means a coding scheme which has rate non-negligibly far from channel capacity.

In this paper, we define a specific "hard" channel. With respect to this channel, we derive lower bounds on the drop in mutual information as a function of the channel input alphabet size, q, and the number of output letters of the approximating channel, L. Simply put, the main result of this paper is that for moderate values of q, a modest drop in mutual information translates into the requirement that L be unreasonably large, in the general case. The result seems interesting by itself: there exists a channel which is hard to approximate by any reasonably sized quantization of the output. The implication for polar codes is: if we were to construct a polar code for transmission over this "hard" channel, and were to go about it by the above discussed method of approximating intermediate channels by degraded channels, we would inevitably construct a code with rate significantly lower than the capacity of the channel, if q is moderate and L is not allowed to be prohibitively large.

It seems to be common knowledge that constructing capacity achieving LDPC or polar codes for channels with moderate input alphabet sizes is generally hard. To quote, for example, [17]:

> Unlike binary LDPC codes, the problem of finding an efficient algorithm for computing density evolution for nonbinary LDPC codes remains open. This is a result of the fact that the messages transferred in nonbinary belief-propagation are multidimensional vectors rather than scalar values. Just storing the density of a non-scalar random variable requires an amount of memory that is exponential in the alphabet size. Nevertheless, we show that approximation using surrogates is very much possible.

To recap, this paper is an attempt to vindicate and quantify this hardness. We will do so by analyzing a specific problem instance — a specific underlying channel.

The structure of this paper is as follows. Section II introduces the main result of the paper, after stating the needed notation. Section III explains the implications of the result to the hardness of constructing polar codes and LPDC codes. Section IV contains a specialization of Hölder's defect formula to our setting. Section V defines and analyzes the previously discussed channel. Section VI concludes the paper, and shortly discusses a probabilistic method of constructing polar codes.

II. Notation and problem statement

We denote a channel by $W\colon \mathcal{X} \to \mathcal{Y}$. The probability of receiving $y \in \mathcal{Y}$ given that $x \in \mathcal{X}$ was transmitted over W is denoted $W(y|x)$. All our channels will be defined over a finite input alphabet $\mathcal{X}$, with size $q = |\mathcal{X}|$. Unless specifically stated otherwise, all channels will have a finite output alphabet, denoted $\mathrm{out}(W) = \mathcal{Y}$. Thus, the channel output alphabet size is denoted $|\mathrm{out}(W)|$.

We will eventually deal with a specific channel, which turns out to be symmetric (as defined in [18, page 94]). In addition, the input distribution we will ultimately assign to this channel turns out to be uniform. However, we would like to be as general as possible wherever appropriate. Thus, unless specifically stated otherwise, we *will not* assume that a generic channel W is symmetric. Each channel will typically have a corresponding input distribution, denoted $P_X = P_X^{(W)}$. Note that P_X *need not* necessarily be uniform and *need not* necessarily be the input distribution achieving the capacity of W. We denote the random variables corresponding to the input and output of W by $X = X^{(W)}$ and $Y = Y^{(W)}$, respectively. The distribution of Y is denoted $P_Y = P_Y^{(W)}$. That is, for $y \in \mathcal{Y}$,

$$P_Y(y) = \sum_{x \in \mathcal{X}} P_X(x) W(y|x) \; .$$

The mutual information between X and Y is denoted as

$$I(W) = I(X;Y) \; ,$$

and is henceforth measured in nats. That is, all logarithms henceforth are natural. We stress that, in general, $I(W)$ *does not* equal the capacity of W.

We say that a channel $Q\colon \mathcal{X} \to \mathcal{Z}$ is (stochastically) degraded with respect to $W\colon \mathcal{X} \to \mathcal{Y}$ if there exists a channel $\Phi\colon \mathcal{Y} \to \mathcal{Z}$ such that the concatenation of Φ to W yields Q. Namely, for all $x \in \mathcal{X}$ and $z \in \mathcal{Z}$,

$$Q(z|x) = \sum_{y \in \mathcal{Y}} W(y|x)\Phi(z|y) \; . \tag{1}$$

We denote Q being degraded with respect to W as $Q \prec W$.

For input alphabet size $q = |\mathcal{X}|$ and specified output alphabet size L, define the *degrading cost* as

$$\mathrm{DC}(q, L) \triangleq \sup_{W, P_X} \min_{\substack{Q \,:\, Q \prec W, \\ |\mathrm{out}(Q)| \le L}} \left(I(W) - I(Q)\right) \; . \tag{2}$$

Namely, both W and Q range over channels with input alphabet $\mathcal{X}$ such that $|\mathcal{X}| = q$; both channels share the same input distribution P_X, which we optimize over; the channel Q is degraded with respect to W; both channels have finite output alphabets and the size of the output alphabet of Q is at most L; we calculate the drop in mutual information incurred by degrading W to Q, for the "hardest" channel W, the "hardest" corresponding input distribution P_X, and the corresponding best approximation Q.

Note that the above explanation of (2) is a bit off, since the outer qualifier is "sup", not "max". Namely, we might need to consider a sequence of channels W and input distributions P_X. Note however that the inner qualifier is a "min", and not an "inf". This is justified by the following claim, which is taken from [11, Lemma 1].

Claim 1: Let $W\colon \mathcal{X} \to \mathcal{Y}$ and P_X be given. Let $L \geq 1$ be a specified integer for which $|\mathcal{Y}| \geq L$. Then,

$$\inf_{\substack{Q\,:\,Q \prec W,\\ |\mathrm{out}(Q)| \leq L}} (I(W) - I(Q))$$

is attained by a channel $Q\colon \mathcal{X} \to \mathcal{Z}$ for which it holds that $|\mathrm{out}(Q)| = L$ and

$$Q(z|x) = \sum_{y \in \mathcal{Y}} W(y|x)\Phi(z|y) \;, \qquad \Phi(z|y) \in \{0,1\} \;,$$
$$\sum_{z \in \mathcal{Z}} \Phi(z|y) = 1 \;.$$

Namely, Q is gotten from W by defining a partition $(A_i)_{i=1}^L$ of $\mathcal{Y}$ and mapping with probability 1 all symbols in A_i to a single symbol $z_i \in \mathcal{Z}$, where $\mathcal{Z} = \{z_i\}_{i=1}^L$.

In [12], an upper bound on $\mathrm{DC}(q, L)$ is derived. Specifically,

$$\mathrm{DC}(q, L) \leq 2q \cdot \left(\frac{1}{L}\right)^{1/q} \;.$$

The above has been recently sharpened [13, Lemma 8] to

$$\mathrm{DC}(q, L) \leq 2 \cdot q^{1+\frac{2}{q-1}} \cdot \left(\frac{1}{L}\right)^{1/(q-1)} \;.$$

These bounds are constructive and stem from a specific quantizing algorithm. Specifically, the algorithm is given as input the channel W, the corresponding input distribution P_X, and an upper bound on the output alphabet size, L. Note that for a fixed input alphabet size q and a target difference ϵ such that $\mathrm{DC}(q, L) \leq \epsilon$, the above implies that we take L proportional to $(1/\epsilon)^{q-1}$. That is, for moderate values of q, the required output alphabet size grows very rapidly in $1/\epsilon$. Because of this, [12] explicitly states that the algorithm can be considered practical only for small values of q.

We now quote our main result: a lower bound on $\mathrm{DC}(q, L)$. Let σ_{q-1} be the constant for which the volume of a ball in $\mathbb{R}^{q-1}$ of radius r is $\sigma_{q-1} r^{q-1}$. Namely,

$$\sigma_{q-1} = \frac{\pi^{\frac{q-1}{2}}}{\Gamma(\frac{q-1}{2} + 1)} \;,$$

where Γ is the Gamma function. That is, for an integer $n \geq 1$,

$$\Gamma(n) = (n-1)! \;, \quad \Gamma\left(n + \frac{1}{2}\right) = \frac{(2n)!}{4^n n!}\sqrt{\pi} \;.$$

Theorem 2: Let q and L be specified. Then,

$$\mathrm{DC}(q, L) \geq \frac{q-1}{2(q+1)} \cdot \left(\frac{1}{\sigma_{q-1} \cdot (q-1)!}\right)^{\frac{2}{q-1}} \cdot \left(\frac{1}{L}\right)^{\frac{2}{q-1}} \;. \qquad (3)$$

The above bound is attained in the limit for a sequence of symmetric channels, each having a corresponding input distribution which is uniform.

The consequences of this theorem in the context of code construction will be elaborated on in the next section. However, one immediate consequence is a vindication of sorts for the algorithm presented in [12]. That is, for q fixed, we deduce from the theorem that the optimal degrading algorithm must take the output alphabet size L at least proportional to $(1/\epsilon)^{(q-1)/2}$, where ϵ is the designed drop in mutual information. That is, the adverse effect of L growing rapidly with $1/\epsilon$ is an inherent property of the problem, and is not the consequence of a poor implementation. For a numerical example, take $q = 16$ and $\epsilon = 10^{-5}$. The theorem states that the optimal degrading algorithm must allow for a target output alphabet size $L \approx 10^{23}$. This number is, for all intents and purposes, intractable.

We note that the term multiplying $(1/L)^{2/(q-1)}$ in (3) can be simplified by Stirling's approximation. The result is that

$$\mathrm{DC}(q, L) \geq (1 - O(1/q)) \cdot \left(\frac{e}{4\pi(q+1)}\right) \cdot \left(\frac{1}{L}\right)^{\frac{2}{q-1}} \;.$$

Note that the RHS of the above is eventually decreasing in q, for L fixed. However, it must be the case that $\mathrm{DC}(q, L)$ is increasing in q (to see this, note that the input distribution can give a probability of 0 to some input symbols). Thus, we conclude that our bound is not tight.

III. Implications for code construction

We now explain the relevance of our result to the construction of both polar codes and LDPC codes. In both cases, a "hard" underlying channel is used, with a corresponding input distribution that is uniform. Let us explain: for q and L fixed, and for a uniform input distribution, we say that a channel is hard if the drop in mutual information incurred by degrading it to a channel with at most L output letters is, say, at least half of the RHS of (3). Theorem 2 assures us that such hard channels exist. Put another way, the crucial point we will make use of is that for a hard channel, the drop in mutual information is at least proportional to $(1/L)^{2/(q-1)}$.

A. Polar codes

As explained in the introduction, the current methods of constructing polar codes for symmetric channels involve approximating the intermediate channels by channels with a manageable output alphabet size. Specifically, the underlying channel — the channel over which the codeword is transmitted — is approximated by degradation before any polarization operation is applied. Now, for q fixed and L a parameter, consider an underlying hard channel, as defined above. Denote the underlying channel as W, and let the result of the initial degrading approximation be denoted by Q.

The key point to note is that the construction algorithm cannot distinguish between W and Q. That is, consider two runs of the construction algorithm, one in which the underlying channel is W and another in which the underlying channel is Q. In the first case, the initial degradation produces Q from W. In the second case, the initial degradation simply returns Q, since the output alphabet size is at most L, and thus no reduction of output alphabet is needed. Thus, the rate of the code constructed cannot be greater than the symmetric capacity of Q, which is at most $I(W) - \epsilon$. We can of course make ϵ arbitrarily small. However, this would necessitate an L at least proportional to $(1/\epsilon)^{(q-1)/2}$. For rather modest values of q and ϵ, this is intractable.

B. LDPC codes

The standard way of designing an LDPC code for a specified underlying channel is by applying the density evolution algorithm [7, Section 4.4]. To simplify to our needs, density evolution performs a series of channel transformations on the underlying channel, which are a function of the degree distribution of the code ensemble considered. Exactly as in the polar coding setting, these transformations increase the output alphabet size to intractable sizes. Thus, in practice, the channels are approximated. If we assume that the approximation is degrading — and it typically is — the rest of the argument is now essentially a repetition of the argument above. In brief, consider an LDPC code designed for a hard channel W. After the first degrading operation, a channel Q is gotten. The result produced by the algorithm when W is the underlying channel must equal the result produced when Q is the underlying channel. Thus, an ensemble with rate above that of the symmetric capacity of Q will necessarily be reported as "bad" with respect to both W and Q. Reducing the mutual information between W and Q is intractably costly for moderate parameter choices.

IV. Preliminary lemmas

As a consequence of the data processing inequality, if Q is degraded with respect to W, then $I(W) - I(Q) \geq 0$. In this section, we derive a tighter lower bound on the difference. To that end, let us first define $\eta(p)$ as

$$\eta(p) = -p \cdot \ln p \; , \quad 0 \leq p \leq 1 \; ,$$

where $\eta(0) = 0$. Next, for a probability vector $\mathbf{p} = (p_x)_{x\in\mathcal{X}}$, define

$$h(\mathbf{p}) = \sum_{x\in\mathcal{X}} -p_x \cdot \ln p_x = \sum_{x\in\mathcal{X}} \eta(p_x) \; .$$

For $A = \{y_1, y_2, \ldots, y_t\} \subseteq \mathcal{Y}$, define the quantity $\Delta(A)$ as the decrease in mutual information resulting from merging all symbols in A into a single symbol in Q. Namely, define

$$\Delta(A) \triangleq \pi \left(h\left[\sum_{j=1}^{t} \theta_j \mathbf{p}^{(j)}\right] - \left(\sum_{j=1}^{t} \theta_j h\big[\mathbf{p}^{(j)}\big]\right)\right) \; , \quad (4)$$

where

$$\pi = \sum_{y\in A} P_Y(y) \; , \quad \theta_j = P_Y(y_j)/\pi \; , \quad (5)$$

and

$$\mathbf{p}^{(j)} = (P(X = x|Y = y_j))_{x\in\mathcal{X}} \; . \quad (6)$$

The following claim is easily derived.

Claim 3: Let W, Q, P_X, L, and $(A_i)_{i=1}^L$ be as in Claim 1. Then,

$$I(W) - I(Q) = \sum_{i=1}^{L} \Delta(A_i) \; . \quad (7)$$

Although the drop in mutual information is easily described, we were not able to analyze and manipulate it directly. We now aim for a bound which is more amenable to analysis. As mentioned, by the concavity of h and Jensen's inequality, we deduce that $\Delta(A_i) \geq 0$. Namely, data processing reduces mutual information. We will shortly make use of the fact that h is strongly concave in order to derive a sharper lower bound. To that end, we now state Hölder's defect formula [19] (see [20, Page 94] for an accessible reference).

As is customary, we will phrase Hölder's defect formula for $\cup$-convex functions, although we will later apply it to h which is $\cap$-concave. We remind the reader that for twice differentiable $\cup$-convex functions, $f\colon D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$, the Hessian of f, denoted

$$\nabla^2 f(\alpha) = \left(\frac{\partial^2 f(\alpha)}{\partial \alpha_i \partial \alpha_j}\right)_{i,j} \; ,$$

is positive semidefinite on the interior of D [21, page 71]. We denote the smallest eigenvalue of $\nabla^2 f(\alpha)$ by $\lambda_{\min}(\nabla^2 f(\alpha))$.

Lemma 4: Let $f(\alpha)\colon D \to \mathbb{R}$ be a twice differentiable convex function defined over a convex domain $D \subseteq \mathbb{R}^n$. Let $m \geq 0$ be such that for all α in the interior of D,

$$m \leq \lambda_{\min}(\nabla^2 f(\alpha))$$

Fix $(\alpha_j)_{j=1}^t \in D$ and let $(\theta_j)_{j=1}^t$ be non-negative coefficients summing to 1. Denote

$$\overline{\alpha} = \sum_{j=1}^{t} \theta_j \alpha_j$$

and

$$\delta^2 = \sum_{j=1}^{t} \theta_j \left\| \alpha_j - \overline{\alpha} \right\|_2^2 = \frac{1}{2} \sum_{j=1}^{t} \sum_{k=1}^{t} \theta_j \theta_k \left\| \alpha_j - \alpha_k \right\|_2^2$$

Then,

$$\sum_{j=1}^{t} \theta_j f[\alpha_j] - f[\sum_j \theta_j \alpha_j] \geq \frac{1}{2} m \delta^2 \; .$$

Proof: Let Λ be a diagonal matrix having all entries equal to m. By definition of m, we have that the function $g(\alpha) = f(\alpha) - \frac{1}{2}\alpha^T \Lambda \alpha$ has a positive semidefinite Hessian for all $\alpha \in D$. Thus, by Jensen's inequality,

$$\sum_i \theta_i g[\alpha_i] - g[\sum_i \theta_i \alpha_i] \geq 0 \; .$$

Replacing $g(\alpha)$ in the above expression by $f(\alpha) - \frac{m}{2}\alpha^T \alpha$ and rearranging yields the required result. ∎

We now apply Hölder's defect formula in order to bound $\Delta(A)$. For $A = \{y_1, y_2, \ldots, y_t\} \subseteq \mathcal{Y}$, define

$$\tilde{\Delta}(A) \triangleq \frac{\pi}{2} \sum_{j=1}^{t} \theta_j \left\| \mathbf{p}^{(j)} - \bar{\mathbf{p}} \right\|_2^2 = \frac{\pi}{4} \sum_{j=1}^{t} \sum_{k=1}^{t} \theta_j \theta_k \left\| \mathbf{p}^{(j)} - \mathbf{p}^{(k)} \right\|_2^2 , \quad (8)$$

where π and θ_j are as in (5), $\mathbf{p}^{(j)}$ is as defined in (6), and

$$\bar{\mathbf{p}} = \sum_{j=1}^{t} \theta_j \mathbf{p}^{(j)} .$$

The following is a simple corollary of Lemma 4

Corollary 5: Let W, Q, P_X, L, and $(A_i)_{i=1}^{L}$ be as in Claim 1. Then, for all $1 \le i \le L$,

$$\Delta(A_i) \ge \tilde{\Delta}(A_i) . \quad (9)$$

Thus,

$$I(W) - I(Q) \ge \sum_{i=1}^{L} \tilde{\Delta}(A_i) . \quad (10)$$

Proof: The second inequality follows from the first inequality and (7). We now prove the first inequality. Let $D = [0,1]^n$, the set of vectors of length n having each entry between 0 and 1. Note that h is indeed twice differentiable on D. Since the second derivative of η is $\eta''(p) = -1/p$, we conclude $\lambda_{\min}(-h(\mathbf{p})) \ge 1$ for all $\mathbf{p}$ in the interior $(0,1)^n$. That is, we take $m = 1$ in Lemma 4. Since h is continuous on D, our result follows by Lemma 4 and standard limiting arguments. ■

V. Bounding the degrading cost

We now turn to bounding the degrading cost. To do this, we define a channel W for which we will prove a lower bound on the cost of degrading.

A. *The channel* W

For a specified integer $M \ge 1$, we now define the channel $\mathsf{W} = \mathsf{W}_M$, where $\mathsf{W}\colon \mathcal{X} \to \mathcal{Y}$. The input alphabet is $\mathcal{X} = \{1, 2, \ldots, q\}$, of size $|\mathcal{X}| = q$. The output alphabet consists of vectors of length q with integer entries, defined as follows:

$$\mathcal{Y} = \left\{ \langle j_1, j_2, \ldots, j_q \rangle : j_1, j_2, \ldots, j_q \ge 0 , \quad \sum_{x=1}^{q} j_x = M \right\} . \quad (11)$$

The channel transition probabilities are given by

$$\mathsf{W}(\langle j_1, j_2, \ldots, j_q \rangle | x) = \frac{q \cdot j_x}{M \binom{M+q-1}{q-1}} .$$

Lemma 6: The above defined W is a valid channel with output alphabet size

$$|\mathrm{out}(\mathsf{W})| = \binom{M+q-1}{q-1} . \quad (12)$$

Proof: The binomial expression for the output alphabet size follows by noting that we are essentially dealing with an instance of "combinations with repetitions" [22, Page 15]. Obviously, the probabilities are non-negative. It remains to show that for all $x \in \mathcal{X}$,

$$\sum_{\langle j_1, j_2, \ldots, j_q \rangle \in \mathcal{Y}} \frac{q \cdot j_x}{M \binom{M+q-1}{q-1}} = 1 .$$

Since the above is independent of x, we can equivalently show that

$$\sum_{\langle j_1, j_2, \ldots, j_q \rangle \in \mathcal{Y}} \frac{q \cdot (j_1 + j_2 + \cdots + j_q)}{M \binom{M+q-1}{q-1}} = q .$$

By the definition of $\mathcal{Y}$ in (11), the numerator above equals $q \cdot M$. Since we have already proved (12), the result follows. ■

Recall the definition of symmetry in [18, page 94]: Let $W : \mathcal{X} \to \mathcal{Y}$ be a channel. Define the probability matrix associated with W as a matrix with rows indexed by $\mathcal{X}$ and columns by $\mathcal{Y}$ such that entry $(x, y) \in \mathcal{X} \times \mathcal{Y}$ equals $W(y|x)$. The channel W is symmetric if the output alphabet can be partitioned into sets, and the following holds: for each set, the corresponding submatrix is such that every row is a permutation of the first row and every column is a permutation of the first column.

Lemma 7: The above defined W is a symmetric channel.

Proof: Define the partition so that two output letters, $\langle j_1, j_2, \ldots, j_q \rangle$ and $\langle j'_1, j'_2, \ldots, j'_q \rangle$, are in the same set if there exists a permutation $\pi : \mathcal{X} \to \mathcal{X}$ such that $j_x = j'_{\pi(x)}$, for all $x \in \mathcal{X}$. ■

Since W is symmetric, it follows from [18, Theorem 4.5.2] that the capacity achieving distribution is the uniform distribution. Thus, we take the corresponding input distribution as uniform. Namely, for all $x \in \mathcal{X}$,

$$P(X = x) = \frac{1}{q} .$$

As a result, all output letters are equally likely (the proof is similar to that of Lemma 6).

Denote the vector of a posteriori probabilities corresponding to $\langle j_1, j_2, \ldots, j_q \rangle$ as

$$\mathbf{p}(j_1, j_2, \ldots, j_q) = (\ P(X = x | Y = \langle j_1, j_2, \ldots, j_q \rangle)\ )_{x=1}^{q} .$$

A short calculation gives

$$\mathbf{p}(j_1, j_2, \ldots, j_q) = \left(\frac{j_1}{M}, \frac{j_2}{M}, \ldots, \frac{j_q}{M} \right) . \quad (13)$$

In light of the above, let us define the shorthand

$$\langle j_1, j_2, \ldots, j_q \rangle \triangleq (j_1/M, j_2/M, \ldots, j_q/M) .$$

With this shorthand in place, the label of each output letter $\langle j_1, j_2, \ldots, j_q \rangle \in \mathcal{Y}$ is the corresponding a posteriori probability vector $\mathbf{p}(j_1, j_2, \ldots, j_q)$. Thus, we gain a simple expression for $\tilde{\Delta}(A)$. Namely, for $A \subseteq \mathcal{Y}$,

$$\tilde{\Delta}(A) = \frac{1}{2\binom{M+q-1}{q-1}} \sum_{\mathbf{p} \in A} \|\mathbf{p} - \bar{\mathbf{p}}\|_2^2 , \quad \bar{\mathbf{p}} = \sum_{\mathbf{p} \in A} \frac{1}{|A|} \mathbf{p} .$$

We remark in passing that as $M \to \infty$, W "converges" to the channel $\mathcal{W}_q\colon \mathcal{X} \to [0,1]^q$ which we now define.

Independently of the input, the channel first picks a vector $\boldsymbol{\varphi} = (\varphi_1, \varphi_2, \ldots, \varphi_q)$ according to the Dirichlet distribution $D(1, 1, \ldots, 1)$. That is, $(\varphi_1, \varphi_2, \ldots, \varphi_q)$ is chosen uniformly from all possible probability vectors of length q. Then, still independently of the input, an index $1 \leq i \leq q$ is picked according to this probability vector. That is, i is picked with probability φ_i. For channel input x, we now simply swap entry i and entry x in the vector $(\varphi_1, \varphi_2, \ldots, \varphi_q)$, or do nothing if i equals x. The resulting vector, which we denote $\hat{\boldsymbol{\varphi}}$, is the output of the channel. We now show that all output vectors are equally likely, and the posterior probability of x give $\hat{\boldsymbol{\varphi}}$ is simply $\hat{\varphi}_x$.

Recall that the probability density function of the pre-swap vector is constant, and denote it by g. Nevertheless, to aid in the explanation, let us abuse notation and also denote the probability density function of $\boldsymbol{\varphi}$ by the (constant) function $g(\boldsymbol{\varphi}) = g$. Next, denote by $f(\hat{\boldsymbol{\varphi}}|x)$ the conditional probability density function of the output $\hat{\boldsymbol{\varphi}}$, given that the input was x. By definition,

$$f(\hat{\boldsymbol{\varphi}}|x) = \sum_{i=1}^{q} g(\sigma_{i,x}[\hat{\boldsymbol{\varphi}}]) \cdot \hat{\varphi}_x = q \cdot g \cdot \hat{\varphi}_x \ ,$$

where $\sigma_{i,x}[\boldsymbol{\varphi}]$ is the vector gotten by swapping entries i and x of $\hat{\boldsymbol{\varphi}}$. We now show that for a uniform input distribution, all output letters are equally likely. Indeed, the probability density function of an output vector $\hat{\boldsymbol{\varphi}}$ equals

$$\sum_{x=1}^{n} p(x) f(\hat{\boldsymbol{\varphi}}|x) = \sum_{x=1}^{n} \frac{1}{q} \cdot q \cdot g \cdot \hat{\varphi}_x = g \sum_{x=1}^{n} \hat{\varphi}_x = g \ ,$$

a constant. The last equality follows since the entries of the pre-swapped vector $\boldsymbol{\varphi}$ sum to 1, and thus this is also the case for $\hat{\boldsymbol{\varphi}}$. Next, we derive the posterior probability of x given $\hat{\boldsymbol{\varphi}}$. By Bayes' theorem,

$$p(x|\hat{\boldsymbol{\varphi}}) = \frac{p(x) f(\hat{\boldsymbol{\varphi}}|x)}{\sum_{i=1}^{n} p(i) f(\hat{\boldsymbol{\varphi}}|i)} = \frac{\frac{1}{q} \cdot q \cdot g \cdot \hat{\varphi}_x}{\sum_{i=1}^{n} \frac{1}{q} \cdot q \cdot g \cdot \hat{\varphi}_i} = \frac{\hat{\varphi}_x}{\sum_{i=1}^{n} \hat{\varphi}_i} = \hat{\varphi}_x \ ,$$

where we have again used the fact that the entries of $\boldsymbol{\varphi}'$ sum to 1.

Note that instead of swapping entries i and x to get from $\boldsymbol{\varphi}$ to $\hat{\boldsymbol{\varphi}}$, we could have instead cyclically rotated $\boldsymbol{\varphi}$ such that entry i is now entry x. Apart from defining $\sigma_{i,x}$ as the inverse rotation, the above derivations are valid for this version as well.

B. Optimizing A'

Our aim is to find a lower bound on $\tilde{\Delta}(A)$, where $A \subseteq \mathcal{Y}$ is constrained to have a size $|A| = t$. Recalling (13), note that all output letters $\mathbf{p} = (p_x)_{x=1}^{q} \in \mathcal{Y}$ must satisfy the following three properties.

1) All entries p_x are of the form j_x/M, where j_x is an integer.
2) All entries p_x sum to 1.
3) All entries p_x are non-negative.

Since all entries must sum to 1 by property 2, entry p_q is redundant. Thus, for a given $\mathbf{p} \in \mathcal{Y}$, denote by $\mathbf{p}'$ the first $q-1$ coordinates of $\mathbf{p}$. Let A' be the set one gets by applying this puncturing operation to each element of A. Denote

$$\tilde{\Delta}(A') \triangleq \frac{1}{2\binom{M+q-1}{q-1}} \sum_{\mathbf{p}' \in A'} \|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 \ , \tag{14}$$

One easily shows that

$$\tilde{\Delta}(A') \leq \tilde{\Delta}(A) \ , \tag{15}$$

thus a lower bound on $\tilde{\Delta}(A')$ is also a lower bound on $\tilde{\Delta}(A)$.

In order to find a lower bound on $\tilde{\Delta}(A')$ we relax constraint 3 above. Namely, a set A' with elements $\mathbf{p}'$ will henceforth mean a set for which each element $\mathbf{p}' = (p_x)_{x=1}^{q-1}$ has entries of the form $p_x = j/M$, and each such entry is *not* required to be non-negative. Our revised aim is to find a lower bound on $\tilde{\Delta}(A')$ where A' holds elements as just defined and is constrained to have size t. The simplification enables us to give a characterization of the optimal A'. Informally, it is a ball, up to irregularities on the boundary.

Lemma 8: Let $t > 0$ be a given integer. Let A' be the set of size $|A'| = t$ for which $\tilde{\Delta}(A')$ is minimized. Denote by $\bar{\mathbf{p}}'$ the mean of all elements of A'. Then, A' has a critical radius r. Namely, all $\mathbf{p}'$ for which $\|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 < r^2$ are in A'; all $\mathbf{p}'$ for which $\|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 > r^2$ are not in A'; there exists $\mathbf{p}' \in A'$ for which $\|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 = r^2$.

Proof: We start by considering a general A'. Suppose $\mathbf{p}'(1) \in A'$ is such that $r^2 = \|\mathbf{p}'(1) - \bar{\mathbf{p}}'\|_2^2$. Next, suppose that there is a $\mathbf{p}'(2) \notin A'$ such that $\|\mathbf{p}'(2) - \bar{\mathbf{p}}'\|_2^2 < r^2$. Then, for

$$B' = A' \cup \{\mathbf{p}'(2)\} \setminus \{\mathbf{p}'(1)\} \ , \quad \tilde{\Delta}(B') < \tilde{\Delta}(A') \ .$$

To see this, first note that

$$\sum_{\mathbf{p}' \in B'} \|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 < \sum_{\mathbf{p}' \in A'} \|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 \ . \tag{16}$$

Next, note that the RHS of (16) is $\tilde{\Delta}(A')$, but the LHS is *not* $\tilde{\Delta}(B')$. Namely, $\bar{\mathbf{p}}'$ is the mean of the vectors in A' but is not the mean of the vectors in B'. However, $\sum_{\mathbf{p}' \in B'} \|\mathbf{p}' - \mathbf{u}'\|_2^2$ is minimized for $\mathbf{u}'$ equal to the mean of the vectors in B' (to see this, differentiate the sum with respect to every coordinate of $\mathbf{u}'$). Thus, the LHS of (16) is at least $\tilde{\Delta}(B')$ while the RHS equals $\tilde{\Delta}(A')$.

The operation of transforming A' into B' as above can be applied repeatedly, and must terminate after a finite number of steps. To see this, note that the sum $\sum_{\mathbf{p}' \in A'} \|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2$ is constantly decreasing, and so is upper bounded by the initial sum. Therefore, one can bound the maximum distance between any two points in A'. Since the sum is invariant to translations, we can always translate A' such that its members are contained in a suitably large hypercube (the translation will preserve the $1/M$ grid property). The number of ways to distribute $|A'|$ grid points inside the hypercube is finite. Since the sum is strictly decreasing and non-negative, the number of steps is finite. The ultimate termination implies a critical r as well as the existence of an optimal A'. ■

Recall that a ball of radius r in $\mathbb{R}^{q-1}$ has volume $\sigma_{q-1}r^{q-1}$, where σ_{q-1} is a well known constant [23, Page 411]. Given a set A', we define the volume of A' as

$$\mathrm{Vol}(A') \triangleq \frac{|A'|}{M^{q-1}} .$$

For optimal A' as above, the following lemma approximates $\mathrm{Vol}(A')$ by the volume of a corresponding ball.

Lemma 9: Let A' be a set of size t for which $\tilde{\Delta}(A')$ is minimized. Let the critical radius be r and assume that $r \leq 3$. Then,

$$\mathrm{Vol}(A') = \sigma_{q-1}r^{q-1} + \epsilon_{q-1}(t) .$$

The error term $\epsilon_{q-1}(t)$ is bounded from both above and below by functions of M alone (*not* of t) that are $o(1)$ (decay to 0 as $M \to \infty$).

Proof: Let $\delta\colon \mathbb{R}^{q-1} \to \{0,1\}$ be the indicator function of a ball with radius r centered at $\bar{\mathbf{p}}'$. That is,

$$\delta(\mathbf{p}') = \begin{cases} 1 & \|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 \leq r^2 \\ 0 & \text{otherwise} . \end{cases}$$

Note that 1) δ is a bounded function and 2) the measure of points for which δ is not continuous is zero (the boundary of a ball has no volume). Thus, δ is Riemann integrable [24, Theorem 14.5].

Consider the set Ψ' which is $[-r,r]^{q-1}$ shifted by $\bar{\mathbf{p}}'$. Since Ψ' contains the above ball, the integral of δ over Ψ' must equal $\sigma_{q-1}r^{q-1}$. We now show a specific Riemann sum [24, Definition 14.2] which must converge to this integral. Consider a partition of Ψ' into cubes of side length $1/M$, where each cube center is of the form $(j_1/M, j_2/M, \ldots, j_{q-1}/M)$ and the j_x are integers (the fact that cubes at the edge of Ψ' are of volume less than $1/M^{q-1}$ is immaterial). Define $[\mathbf{p}' \in A']$ as 1 if the condition $\mathbf{p}' \in A'$ holds and 0 otherwise. We claim that the following is a Riemann sum of δ over Ψ' with respect to the above partition.

$$\sum_{\mathbf{p}'=(j_1/M, j_2/M, \ldots, j_{q-1}/M) \in \Psi'} \frac{1}{M^{q-1}} [\mathbf{p}' \in A']$$

To see this, recall that A' has critical radius r.

The absolute value of the difference between the above sum and $\sigma_{q-1}r^{q-1}$ can be upper bounded by the number of cubes that straddle the ball times their volume $1/M^{q-1}$ (any finer partition will only affect these cubes). Since $r \leq 3$, this quantity must go to zero as M grows, no matter how we let r depend on M. ■

Lemma 10: Let A' be a set of size t for which $\tilde{\Delta}(A')$ is minimized. Let the critical radius be r and assume that $r \leq 3$. Then,

$$\tilde{\Delta}(A') = \frac{(q-1)\cdot(q-1)!}{2(q+1)} \sigma_{q-1} r^{q+1} + \epsilon_{q-1}(t) .$$

The error term $\epsilon_{q-1}(t)$ is bounded from both above and below by functions of M alone (*not* of t) that are $o(1)$ (decay to 0 as $M \to \infty$).

Proof: Let the ball indicator function δ and the bounding set Ψ' be as in the proof of Lemma 9. Consider the sum

$$\sum_{\mathbf{p}'=(j_1/M, j_2/M, \ldots, j_{q-1}/M) \in \Psi'} \frac{\|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 \cdot [\mathbf{p}' \in A']}{M^{q-1}} . \quad (17)$$

On the one hand, by (14), this sum is simply

$$\frac{2\binom{M+q-1}{q-1}}{M^{q-1}} \tilde{\Delta}(A') . \quad (18)$$

On the other hand, (17) is the Riemann sum corresponding to the integral

$$\int_{\Psi'} \|\mathbf{p}' - \bar{\mathbf{p}}'\|_2^2 \, [\mathbf{p}' \in A'] \, d\mathbf{p}' ,$$

with respect to the same partition as was used in the proof of Lemma 9. As before, the sum must converge to the integral, and the convergence rate can be shown to be bounded by expressions which are not a function of t.

All that remains is to calculate the integral. Denote by $\mathrm{ball}_{q-1}(r) \subseteq \mathbb{R}^{q-1}$ the ball centered at the origin with radius r. After translating $\bar{\mathbf{p}}'$ to the origin, the integral becomes

$$\int_{\mathrm{ball}_{q-1}(r)} \left(x_1^2 + x_2^2 + \cdots + x_{q-1}^2\right) \, dx_1 dx_2 \cdots dx_{q-1} = \frac{\sigma_{q-1}\cdot(q-1)\cdot r^{q+1}}{q+1} , \quad (19)$$

where the RHS is derived as follows. After converting the integral to generalized spherical coordinates

$$\begin{aligned} x_1 &= r\cos(\theta_1) , \\ x_2 &= r\sin(\theta_1)\cos(\theta_2) , \\ &\vdots \\ x_{q-2} &= r\sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{q-2})\cos(\theta_{q-1}) , \\ x_{q-1} &= r\sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{q-2})\sin(\theta_{q-1}) , \end{aligned}$$

we get an integrand that is r^2 times the integrand we would have gotten had the original integrand been 1 (this follows by applying the identity $\sin^2\theta + \cos^2\theta = 1$ repeatedly). We know that had that been the case, the integral would have equaled $\sigma_{q-1}r^{q-1}$.

Since (19) must equal the limit of (18), and since the fraction in (18) converges to $2/(q-1)!$, the claim follows. ■

As a corollary to the above three lemmas, we have the following result. The important point to note is that the RHS is convex in $\mathrm{Vol}(A')$.

Corollary 11: Let $t > 0$ be a given integer. Let A' be a set of size t and assume that

$$\max_{\mathbf{p}' \in A'} \|\mathbf{p}' - \bar{\mathbf{p}}'\|_2 \leq 2 . \quad (20)$$

Then,

$$\tilde{\Delta}(A') \geq \frac{(q-1)\cdot(q-1)!}{2(q+1)\cdot(\sigma_{q-1})^{\frac{2}{q-1}}} \cdot \mathrm{Vol}(A')^{\frac{q+1}{q-1}} + o(1) , \quad (21)$$

where the $o(1)$ is a function of M alone and goes to 0 as $M \to \infty$.

Proof: Let B' be the set of size t for which $\tilde{\Delta}(B')$ is minimized. Thus, $\mathrm{Vol}(A') = \mathrm{Vol}(B')$ while $\tilde{\Delta}(B') \leq \tilde{\Delta}(A')$. We conclude that (21) will follow from Lemmas 9 and 10, if we manage to prove that the critical radius of B' is at most

3. We do so next, with the help of (20). We will also assume w.l.o.g. that

$$M \geq \sqrt{q-1} \; . \tag{22}$$

Denote by $\alpha = \bar{\mathbf{p}}'$ and β the centers of A' and B', respectively. Since Vol and $\tilde{\Delta}$ are invariant to the translation of the set, we assume w.l.o.g. that α and β are equal, up to a difference of at most $1/M$ in each coordinate. Thus,

$$\|\alpha - \beta\|_2^2 \leq (q-1)/M^2 \; . \tag{23}$$

Assume to the contrary that the critical radius of B' is greater than 3. We will show that A' is strictly contained in B', a contradiction, since both contain exactly t elements.

First, let us show that A' and B' are distinct, by showing the existence of a $\mathbf{p}'$ such that $\mathbf{p}' \in B'$ but $\mathbf{p}' \notin A'$. Indeed, since the critical radius of B' is greater than 3, there exists a $\mathbf{p}' \in B'$ such that

$$\|\mathbf{p}' - \beta\|_2 > 3 \; . \tag{24}$$

By the triangle inequality,

$$\|\mathbf{p}' - \beta\|_2 \leq \|\mathbf{p}' - \alpha\|_2 + \|\alpha - \beta\|_2 \; . \tag{25}$$

Thus,

$$\|\mathbf{p}' - \alpha\|_2 \geq \|\mathbf{p}' - \beta\|_2 - \|\alpha - \beta\|_2 > 3 - \frac{\sqrt{q-1}}{M} \geq 2 \; ,$$

where the first inequality is a rearrangement of the triangle inequality (25); the second inequality is a consequence of (23) and (24); the third inequality is the result of our assumption on M in (22). We conclude that $\|\mathbf{p}' - \alpha\|_2 > 2$, and from (20) we deduce that $\mathbf{p}' \notin A'$.

Let us now show that A' is contained in B'. Indeed, let $\mathbf{p}' \in A'$, and recall that $\alpha = \bar{\mathbf{p}}'$. Thus, by combining the triangle inequality (25) with (20) and (23) we have that

$$\|\mathbf{p}' - \beta\|_2 \leq 2 + \frac{\sqrt{q-1}}{M} \leq 3 \; ,$$

where the second inequality follows from (22). We deduce that $\mathbf{p}' \in B'$, since the critical radius of B' is greater than 3. Thus, $A' \subseteq B'$. ∎

C. *Bounding* $\mathrm{DC}(q, L)$

We are now in a position to prove Theorem 2. Recall that A_i is the set of output letters in $\mathcal{Y}$ which get mapped to the letter $z_i \in \mathcal{Z}$. Also, recall that A_i' is simply A_i with the last entry dropped from each vector.

Proof of Theorem 2: By combining (2), (10), (15), and (21), we have that as long as condition (20) holds for all A_i', $1 \leq i \leq L$, the degrading cost $\mathrm{DC}(q, L)$ is at least

$$\frac{(q-1)\cdot(q-1)!}{2(q+1)\cdot(\sigma_{q-1})^{\frac{2}{q-1}}} \sum_{i=1}^{L} \mathrm{Vol}(A_i')^{\frac{q+1}{q-1}} + o(1) \; . \tag{26}$$

Recalling that the elements of A are probability vectors, we deduce that condition (20) must indeed hold. Indeed,

$$\|\mathbf{p}' - \bar{\mathbf{p}}'\|_2 \leq \|\mathbf{p}' - \bar{\mathbf{p}}'\|_1 \leq \|\mathbf{p}'\|_1 + \|\bar{\mathbf{p}}'\|_1 \leq 2 \; .$$

The first inequality is a specialization of a standard inequality between L_p norms [25, Theorem 19, page 28]. The second inequality is the triangle inequality. The third inequality follows from $\mathbf{p}'$ and $\bar{\mathbf{p}}'$ being truncated probability vectors.

Next, recall that $\mathrm{Vol}(A_i') = \mathrm{Vol}(A_i)$, and thus

$$\sum_{i=1}^{L} \mathrm{Vol}(A_i') = \frac{|\mathrm{out}(\mathsf{W})|}{M^{q-1}} = \frac{\binom{M+q-1}{q-1}}{M^{q-1}} \; . \tag{27}$$

Note that the RHS converges to $1/(q-1)!$ as $M \to \infty$. By convexity, we have that if we are constrained by (27), then the sum in (26) is lower bounded by setting all $\mathrm{Vol}(A_i')$ equal to the RHS of (27) divided by L. Thus, after taking $M \to \infty$, we get (3). ∎

VI. Conclusions

In this paper, we have shown that there is an inherent loss in capacity incurred when degrading a certain "hard" channel to a new channel with a smaller output alphabet size. Thus, one cannot hope for a general method of constructing polar codes which employs degradation on the one hand, and constructs a code with rate close to the capacity of the channel on the other hand. To be more precise, the previous sentence is true when the input alphabet of the channel is at least moderately large. All of the above is due to the fact that the output alphabet size needed to ensure a given gap from the channel capacity grows exponentially in the input alphabet size. The phenomenon in which the complexity of a problem grows exponentially in the number of dimensions is know as the "curse of dimensionality", a term apparently coined by Bellman [26, page ix] and commonly used in other fields.

One way of overcoming this difficulty is by employing randomization. Namely, for a given synthesized channel, one can repeatedly feed the channel input, produce a corresponding output, and test whether an ML decoder would have produced the correct input, given the output. This allows us to gauge the probability of misdecoding of the channel. Indeed, this method was essentially proposed in the seminal paper [1, Section IX]. That is, we have simplified the discussion in [1, Section IX] by considering the probability of misdecoding and not the Bhattacharyya parameter. For an early use with respect to LDPC codes, see [27].

A typical way of carrying out the above probabilistic construction would be the following. We would like to test whether a specified synthesized channel has probability of misdecoding smaller than a prescribed parameter p. Since the test is probabilistic, there is a probability of the test giving a wrong answer: either returning "true" (a "good" channel) when the correct answer is "false" (a "bad" channel), or the other way around. Since the inclusion of a channel which is too noisy results in a code with poor error correcting performance, we are much more wary of the first type of error. Thus, we would like this error to occur with probability not greater than $\alpha > 0$, a suitably small number. To this end, we set a threshold $\tau = \rho p$, where $0 < \rho < 1$, carry out N decoding trials, and produce the answer "true" only if the number of decoding failures was less than $\tau \cdot N$. Larger values of ρ result in more channels being included on the one hand, and a larger number of trials, N, on the other hand. If we set $\rho = 1/2$ and use the bound [28, page 64, Equation (4.2)], the number of trials is set to

$$N = \left\lceil \frac{3\ln(1/\alpha)}{2p} \right\rceil . \tag{28}$$

One may alternatively employ the more exact bound [28, page 64, Equation (4.1)], valid for all $0 < \rho < 1$.

The biggest merit of the above method is that it is not drastically affected by the input alphabet size q. That is, the channel evaluation time only grows linearly in q. The drawback is that, for α fixed, the number of trials scales like $1/p$. Thus, in contrast with the method in [9], one cannot tractably construct codes with a guarantee (or even a probabilistic guarantee) of decoding error roughly $2^{-\sqrt{n}}$. Such codes exist by [29].

We remark in passing that the grim conclusion of the preceding paragraph can be circumvented — at least in theory — if one employs the following modification. For simplicity of exposition, let us consider the binary input channel. Tracing the proof of [29], we see that there are two phases considered. In the first phase, $m' < m$ channel transformations are carried out. The result is a set of $2^{m'}$ channels, polarized in the sense that most channels have a Bhattacharyya parameter which is fairly close to 0 or fairly close to 1. For the remaining $m - m'$ steps, the proof essentially considers the evolution of the Bhattacharyya parameter. Namely, a plus (minus) transform of a channel with Bhattacharyya parameter Z results in a channel with Bhattacharyya parameters equal to Z^2 (at most $2Z - Z^2$). These bounds suffice in order to show strong polarization.

To recap, if we knew the exact Bhattacharyya parameters of the channels at stage m', we could build a polar code with very low probability of error. Although we do not have an efficient method of calculating the Bhattacharyya parameter *exactly*, we can in fact give a tight enough *upper bound*, with high probability, using the method above. Specifically, a slight modification of the method above allows us to obtain an upper bound on the probability of error of an intermediate channel. Since an upper bound on the probability of error implies an upper bound on the Bhattacharyya parameter [7, Second inequality in (4.65), page 202], we are essentially done.

Acknowledgments: The author thanks Eren Şaşoğlu for his feedback. The comments of the anonymous reviewers have improved and clarified this paper, and are gratefully acknowledged.

References

[1] E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inform. Theory*, vol. 55, pp. 3051–3073, 2009.

[2] E. Şaşoğlu, E. Telatar, and E. Arıkan, “Polarization for arbitrary discrete memoryless channels,” `arXiv:0908.0302v1`, 2009.

[3] R. Mori and T. Tanaka, “Non-binary polar codes using Reed-Solomon codes and algebraic geometry codes,” in *Proc. IEEE Inform. Theory Workshop (ITW'2010)*, Dublin, Ireland, 2010, pp. 1–5.

[4] ——, “Source and channel polarization over finite fields and Reed-Solomon matrices,” *IEEE Trans. Inform. Theory*, vol. 60, pp. 2720–2736, 2014.

[5] N. Presman, O. Shapira, and S. Litsyn, “Polar codes with mixed kernels,” in *Proc. IEEE Int'l Symp. Inform. Theory (ISIT'2011)*, Saint Petersburg, Russia, 2011, pp. 6–10.

[6] E. Şaşoğlu, “Polar codes for discrete alphabets,” in *Proc. IEEE Int'l Symp. Inform. Theory (ISIT'2012)*, Cambridge, Massachusetts, 2012, pp. 2137–21 141.

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

[8] R. Mori and T. Tanaka, “Performance and construction of polar codes on symmetric binary-input memoryless channels,” in *Proc. IEEE Int'l Symp. Inform. Theory (ISIT'2009)*, Seoul, South Korea, 2009, pp. 1496–1500.

[9] I. Tal and A. Vardy, “How to construct polar codes,” *IEEE Trans. Inform. Theory*, vol. 59, pp. 6562–6582, 2013.

[10] R. Pedarsani, S. H. Hassani, I. Tal, and E. Telatar, “On the construction of polar codes,” in *Proc. IEEE Int'l Symp. Inform. Theory (ISIT'2011)*, Saint Petersburg, Russia, 2011, pp. 11–15.

[11] B. M. Kurkoski and H. Yagi, “Quantization of binary-input discrete memoryless channels, with applications to LDPC decoding,” `arXiv:11107.5637v1`, 2011.

[12] I. Tal, A. Sharov, and A. Vardy, “Constructing polar codes for non-binary alphabets and MACs,” in *Proc. IEEE Int'l Symp. Inform. Theory (ISIT'2012)*, Cambridge, Massachusetts, 2012, pp. 2132–2136.

[13] U. Pereg and I. Tal, “Channel upgradation for non-binary input alphabets and MACs,” `arXiv:1308.5793v3`, 2015.

[14] T. C. Gulcu, M. Ye, and A. Barg, “Construction of polar codes for arbitrary discrete memoryless channels,” `arXiv:1603.05736v1`, 2016.

[15] V. Guruswami and A. Velingker, “An entropy sumset inequality and polynomially fast convergence to Shannon capacity over all alphabets,” `arXiv:1411.6993`, 2014.

[16] A. Ghayoori and T. A. Gulliver, “Upgraded approximation of non-binary alphabets for polar code construction,” `arXiv:1304.1790v3`, 2013.

[17] A. Bennatan and D. Burshtein, “Design and analysis of nonbinary ldpc codes for arbitrary discrete-memoryless channels,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 549–583, 2006.

[18] R. G. Gallager, *Information Theory and Reliable Communications*. New York: John Wiley, 1968.

[19] O. Hölder, “Über einen mittelwerthssatz,” *Nachrichten von der Königlichen Gesellschaft der Wissenschaften und der Georg-Augusts-Universität zu Göttingen*, no. 2, pp. 38–47, Jan. 1889.

[20] J. M. Steele, *The Cauchy–Schwarz Master Class*. Cambridge, UK: Cambridge University Press, 2004.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[22] R. P. Stanley, *Enumerative Combinatorics*. Cambridge, UK: Cambridge University Press, 1997, vol. 1.

[23] T. M. Apostol, *Calculus*, 2nd ed. Wiley, 1967, vol. 2.

[24] ——, *Mathematical Analysis*, 2nd ed. Reading, Massachusetts: Addison-Wesley, 1974.

[25] G. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*. Cambridge, UK: Cambridge University Press, 1952.

[26] R. Bellman, *Dynamic Programming*. Princeton, New Jersey: Princeton University Press, 1972.

[27] K. Nakamura, Y. Kabashima, and D. Saad, “Statistical mechanics of low-density parity check error-correcting codes over Galois fields,” *Europhys. Lett.*, vol. 56, pp. 610–616, 2001.

[28] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge, UK: Cambridge University Press, 2005.

[29] E. Arıkan and E. Telatar, “On the rate of channel polarization,” in *Proc. IEEE Int'l Symp. Inform. Theory (ISIT'2009)*, Seoul, South Korea, 2009, pp. 1493–1495.

Ido Tal (S'05–M'08) was born in Haifa, Israel, in 1975. He received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Technion-Israel Institute of Technology, Haifa, Israel, in 1998, 2003 and 2009, respectively. During 2010-2012 he was a postdoctoral scholar at the University of California at San Diego. In 2012 he joined the Electrical Engineering Department at Technion. His research interests include constrained coding and error-control coding.