

Performance of Polar Codes for Channel and Source Coding

Nadine Hussami

AUB, Lebanon, Email: njh03@aub.edu.lb

Satish Babu Korada and Rüdiger Urbanke

EPFL, Switzerland, Email: {satish.korada,ruediger.urbanke}@epfl.ch

Abstract—Polar codes, introduced recently by Arıkan, are the first family of codes known to achieve capacity of symmetric channels using a low complexity successive cancellation decoder. Although these codes, combined with successive cancellation, are optimal in this respect, their finite-length performance is not record breaking. We discuss several techniques through which their finite-length performance can be improved. We also study the performance of these codes in the context of source coding, both lossless and lossy, in the single-user context as well as for distributed applications.

I. INTRODUCTION

Polar codes, recently introduced by Arıkan in [1], are the first provably capacity achieving family of codes for arbitrary symmetric binary-input discrete memoryless channels (B-DMC) with low encoding and decoding complexity. The construction of polar codes is based on the following observation: Let $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Apply the transform $G_2^{\otimes n}$ (where “ $\otimes n$ ” denotes the n^{th} Kronecker power) to a block of $N = 2^n$ bits and transmit the output through independent copies of a symmetric B-DMC, call it W . As n grows large, the channels seen by individual bits (suitably defined in [1]) start *polarizing*: they approach either a noiseless channel or a pure-noise channel, where the fraction of channels becoming noiseless is close to the capacity $I(W)$. In the following, let \bar{u} denote the vector (u_0, \dots, u_{N-1}) and \bar{x} denote the vector (x_0, \dots, x_{N-1}) . Let $\pi : \{0, \dots, N-1\} \rightarrow \{0, \dots, N-1\}$ be the permutation such that if the n -bit binary representation of i is $b_{n-1} \dots b_0$, then $\pi(i) = b_0 \dots b_{n-1}$ (we call this a *bit-reversal*). Let $\text{wt}(i)$ denote the number of ones in the binary expansion of i .

A. Construction of Polar Codes

The channel polarization phenomenon suggests to use the noiseless channels for transmitting information while fixing the symbols transmitted through the noisy ones to a value known both to sender as well as receiver. For symmetric channels we can assume without loss of generality that the fixed positions are set to 0. Since the fraction of channels becoming noiseless tends to $I(W)$, this scheme achieves the capacity of the channel.

In [1] the following alternative interpretation was mentioned; the above procedure can be seen as transmitting a codeword and decoding at the receiver with a successive cancellation (SC) decoding strategy. The specific code which is constructed can be seen as a generalization of the Reed-Muller (RM) codes. Let us briefly discuss the construction of

RM codes. We follow the lead of [2] in which the Kronecker product is used. RM codes are specified by the two parameters n and r and the code is denoted by $\text{RM}(n, r)$. An $\text{RM}(n, r)$ code has block length 2^n and rate $\frac{1}{2^n} \sum_{i=0}^r \binom{n}{i}$. The code is defined through its generator matrix as follows. Compute the Kronecker product $G_2^{\otimes n}$. This gives a $2^n \times 2^n$ matrix. Label the rows of this matrix as $0, \dots, 2^n - 1$. One can check that the weight of the i th row of this matrix is equal to $2^{\text{wt}(i)}$. The generator matrix of the code $\text{RM}(n, r)$ consists of all the rows of $G_2^{\otimes n}$ which have weight at least 2^{n-r} . There are exactly $\sum_{i=0}^r \binom{n}{i}$ such rows. An equivalent way of expressing this is to say that the codewords are of the form $\bar{x} = \bar{u}G_2^{\otimes n}$, where the components u_i of \bar{u} corresponding to the rows of $G_2^{\otimes n}$ of weight less than 2^{n-r} are fixed to 0 and the remaining components contain the “information.” Polar codes differ from RM codes only in the choice of generator vectors \bar{u} , i.e., in the choice of which components of \bar{u} are set to 0. Unlike RM codes, these codes are defined for any dimension $1 \leq k \leq 2^n$. The choice of the generator vectors, as explained in [1], is rather complicated; we therefore do not discuss it here. Following Arıkan, call those components u_i of \bar{u} which are set to 0 “frozen,” and the remaining ones “information” bits. Let the set of frozen bits be denoted by F and the set of information bits be denoted by I . A polar code is then defined as the set of codewords of the form $\bar{x} = \bar{u}G_2^{\otimes n}$, where the bits $i \in F$ are fixed to 0.

Lemma 1 (RM Code as Polar Code): A $\text{RM}(n, r)$ is a polar code of length 2^n with $F = \{u_i : \text{wt}(i) < r\}$.

B. Performance under Successive Cancellation Decoding

In [1] Arıkan considers a low complexity SC decoder. We briefly describe the decoding procedure here. The bits are decoded in the order $\pi(0), \dots, \pi(N-1)$. Let the estimates of the bits be denoted by $\hat{u}_0, \dots, \hat{u}_{N-1}$. If a bit u_i is frozen then $\hat{u}_i = 0$. Otherwise the decoding rule is the following:

$$\hat{u}_{\pi(i)} = \begin{cases} 0, & \text{if } \frac{\Pr(y_0^{N-1} | \hat{u}_{\pi(0)}^{\pi(i-1)}, U_{\pi(i)}=0)}{\Pr(y_0^{N-1} | \hat{u}_{\pi(0)}^{\pi(i-1)}, U_{\pi(i)}=1)} > 1, \\ 1, & \text{otherwise.} \end{cases}$$

Using the factor graph representation between \bar{u} and \bar{x} shown in Figure 2(a), Arıkan showed that this decoder can be implemented with $O(N \log N)$ complexity.

Theorem 2: [3] Let W be any symmetric B-DMC with $I(W) > 0$. Let $R < I(W)$ and $\beta < \frac{1}{2}$ be fixed. Then for $N = 2^n$, $n \geq 0$, the probability of error for polar coding

under SC decoding at block length N and rate R satisfies $P_e(N, R) = o(2^{-N^\beta})$.

C. Optimality of the Exponent

The following lemma characterizes the minimum distance of a polar code.

Lemma 3 (Minimum Distance of Polar Codes): Let I be the set of information bits of a polar code \mathcal{C} . The minimum distance of the code is given by $d_{\min}(\mathcal{C}) = \min_{i \in I} 2^{\text{wt}(i)}$.

Proof: Let $w_{\min} = \min_{i \in I} \text{wt}(i)$. Clearly, d_{\min} cannot be larger than the minimum weight of the rows of the generator matrix. Therefore, $d_{\min} \leq 2^{w_{\min}}$. On the other hand, by adding some extra rows to the generator matrix we cannot increase the minimum distance. In particular, add all the rows of $G_2^{\otimes n}$ with weight at least $2^{w_{\min}}$. The resulting code is $\text{RM}(n, n - w_{\min})$. It is well known that $d_{\min}(\text{RM}(n, r)) = 2^{n-r}$ [2]. Therefore, $d_{\min} \geq d_{\min}(\text{RM}(n, n - w_{\min})) = 2^{w_{\min}}$. ■

We conclude that for any given rate R , if the information bits are picked according to their weight (RM rule), i.e., the $2^n R$ vectors of largest weight, the resulting code has the largest possible minimum distance. The following lemma gives a bound on the best possible minimum distance for any non-trivial rate.

Lemma 4: For any rate $R > 0$ and any choice of information bits, the minimum distance of a code of length 2^n is bounded as $d_{\min} \leq 2^{\frac{n}{2} + c\sqrt{n}}$ for $n > n_o(R)$ and a constant $c = c(R)$.

Proof: Lemma 3 implies that d_{\min} is maximized by choosing the frozen bits according to the RM rule. The matrix $G_2^{\otimes n}$ has $\binom{n}{i}$ rows of weight 2^i . Therefore, $d_{\min} \leq 2^k : \sum_{i=k+1}^n \binom{n}{i} < 2^n R \leq \sum_{i=k}^n \binom{n}{i}$. For $R > \frac{1}{2}$, more than half of the rows are in the generator matrix. Therefore, there is at least one row with weight less than or equal to $2^{\lceil \frac{n}{2} \rceil}$. Consider therefore an R in the range $(0, 1/2]$. Using Stirling's approximation, one can show that for any $R > 0$ at least one row of the generator matrix has weight in the range $[2^{\lceil \frac{n}{2} \rceil - c\sqrt{n}}, 2^{\lceil \frac{n}{2} \rceil + c\sqrt{n}}]$. Using Lemma 3 we conclude the result. ■

Theorem 5: Let $R > 0$ and $\beta > \frac{1}{2}$ be fixed. For any symmetric B-DMC W , and $N = 2^n$, $n \geq n(\beta, R, W)$, the probability of error for polar coding under MAP decoding at block length N and rate R satisfies $P_e(N, R) > 2^{-N^\beta}$.

Proof: For a code with minimum distance d_{\min} , the block error probability is lower bounded by $2^{-Kd_{\min}}$ for some positive constant K , which only depends on the channel. This is easily seen by considering a genie decoder; the genie provides the correct value of all bits except those which differ between the actually transmitted codeword and its minimum distance cousin. Lemma 4 implies that for any $R > 0$, for n large enough, $d_{\min} < \frac{1}{K}N^\beta$ for any $\beta > \frac{1}{2}$. Therefore $P_e(N, R) > 2^{-N^\beta}$ for any $\beta > \frac{1}{2}$. ■

This combined with Theorem 2, implies that the SC decoder achieves performance comparable to the MAP decoder in terms of the order of the exponent.

II. PERFORMANCE UNDER BELIEF PROPAGATION

Theorem 2 does not state what lengths are needed in order to achieve the promised rapid decay in the error probability nor does it specify the involved constants. Indeed, for moderate lengths polar codes under SC decoding are not record breaking. In this section we show various ways to improve the performance of polar codes by considering belief propagation (BP) decoding. BP was already used in [4] to compare the performance of polar codes based on Arıkan's rule and RM rule. For all the simulation points in the plots the 95% confidence intervals are shown. In most cases these confidence intervals are smaller than the point size and are therefore not visible.

A. Successive Decoding as a Particular Instance of BP

For communication over a binary erasure channel (BEC) one can easily show the following.

Lemma 6: Decoding the bit U_i with the SC decoder is equivalent to applying BP with the knowledge of U_0, \dots, U_{i-1} and all other bits unknown (and a uniform prior on them).

We conclude that if we use a standard BP algorithm (such a decoder has access also to the information provided by the frozen bits belonging to U_{i+1}, \dots, U_{N-1}) then it is in general strictly better than a SC decoder. Indeed, it is not hard to construct explicit examples of codewords and erasure patterns to see that this inclusion is strict (BP decoder succeeds but the SC decoder does not). Figure 3 shows the simulation results for the SC, BP and the MAP decoders when transmission takes place over the BEC. As we can see from these simulation results, the performance of the BP decoder lies roughly half way between that of the SC decoder and that of the MAP decoder. For the BEC the *scheduling* of the individual messages is irrelevant to the performance as long as each edge is updated until a fixed point has been reached. For general B-DMCs the performance relies heavily on the specific schedule. We found empirically that a good performance can be achieved by the following schedule. Update the messages of each of the n sections of the trellis from right to left and then from left to right and so on. Each section consists of a collection of Z shaped sub-graphs. We first update the lower horizontal edge, then the diagonal edge, and, finally, the upper horizontal edge of each of these Z sections. In this schedule the information is spread from the variables belonging to one level to its neighboring level. Figure 1 shows the simulation results for the SC decoder and the BP decoder over the binary input additive white Gaussian noise channel (BAWGNC) of capacity $\frac{1}{2}$. Again, we can see a marked improvement of the BP decoder over the SC decoder.

B. Overcomplete Representation: Redundant Trellises

For the polar code of length 2^3 one can check that all the three trellises shown in Figure 2 are valid representations. In fact, for a code of block length 2^n , there exist $n!$ different representations obtained by different permutations of the n layers of connections. Therefore, we can connect the vectors \bar{x} and \bar{u} with any number of these representations and this results

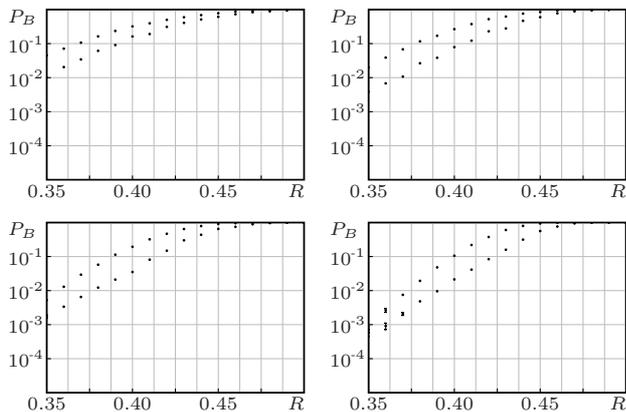


Fig. 1. Comparison of (i) SC and (ii) BP decoder in terms of block error probability, when transmission takes place over the BAWGNC ($\sigma = 0.97865$). The performance curves are shown for $n = 10$ (top left), 11 (top right), 12 (bottom left), and 13 (bottom right).

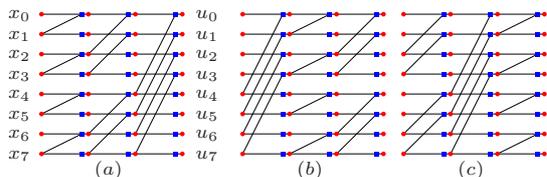


Fig. 2. Figure (a) shows the trellis representation used by Arikan. Figures (b) and (c) show equivalent representations obtained by cyclic shifts of the 3 sections of trellis (a).

in an *overcomplete* representation (similar to the concept used when computing the stopping redundancy of a code [5]). For the BEC any such overcomplete representation only improves the performance of the BP decoder [5]. Further, the decoding complexity scales linearly with the number of different representations used. Keeping the complexity in mind, instead of considering all the $n!$ factorial trellises, we use only the n trellises obtained by cyclic shifts (e.g., see Figure 2). The complexity of this algorithm is $O(N(\log N)^2)$ as compared to $O(N \log N)$ of the SC decoder and BP over one trellis. The performance of the BP decoder is improved significantly by using this overcomplete representation as shown in Figure 3.

We leave a systematic investigation of good schedules and choices of overcomplete representations for general symmetric channels as an interesting open problem.

C. Choice of Frozen Bits

For the BP or MAP decoding algorithm the choice of frozen bits as given by Arikan is not necessarily optimal. In the case of MAP decoding we observe (see Figure 4) that the performance is significantly improved by picking the frozen bits according to the RM rule. This is not a coincidence; d_{\min} is maximized for this choice. This suggests that there might be a rule which is optimized for BP. It is an interesting open question to find such a rule.

III. SOURCE CODING

In this section we show the performance of polar codes in the context of source coding. We consider both lossless and

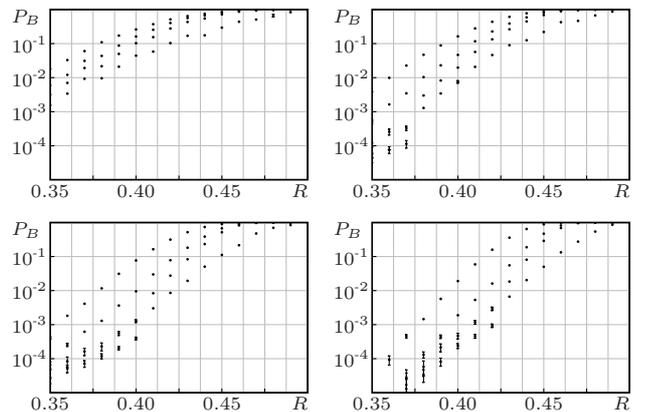


Fig. 3. Comparison of (i) SC, (ii) BP, (iii) BP with multiple trellises, and (iv) MAP in terms of block error probability, when transmission takes place over the BEC ($\epsilon = \frac{1}{2}$). The performance curves are shown for $n = 10$ (top left), 11 (top right), 12 (bottom left), 13 (bottom right).

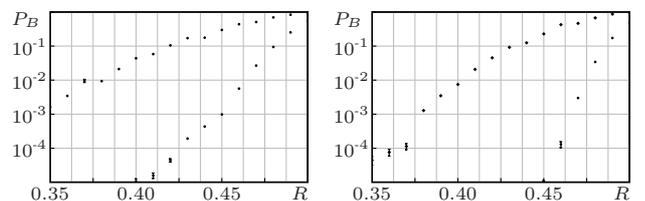


Fig. 4. Comparison of block error probability curves under MAP decoding between codes picked according to Arikan's rule and RM rule. The performance curves are shown for $n = 10$ (left) and 11 (right).

lossy cases and show (in most cases) empirically that they achieve the optimal performance in both cases. Let $\text{Ber}(p)$ denote a Bernoulli source with $\Pr(1) = p$. Let $h_2(\cdot)$ denote the binary entropy function and $h_2^{-1}(\cdot)$ its inverse.

A. Lossless Source Coding

1) *Single User*: The problem of lossless source coding of a $\text{Ber}(p)$ source can be mapped to the channel coding problem over a binary symmetric channel (BSC) as shown in [6], [7]. Let $\bar{x} = (x_0, \dots, x_{N-1})^T$ be a sequence of N i.i.d. realizations of the source. Consider a code of rate R represented by the parity check matrix \mathbf{H} . The vector \bar{x} is encoded by its syndrome $\bar{s} = \mathbf{H}\bar{x}$. The rate of the resulting source code is $1 - R$. The decoding problem is to estimate \bar{x} given the syndrome \bar{s} . This is equivalent to estimating a noise vector in the context of channel coding over $\text{BSC}(p)$. Therefore, if a sequence of codes achieve capacity over $\text{BSC}(p)$, then the corresponding source codes approach a rate $h_2(p)$ with vanishing error probability.

We conclude that polar codes achieve the Shannon bound for lossless compression of a binary memoryless source. Moreover, using the trellis of Figure 2(a), we can compute the syndrome with complexity $O(N \log N)$. The source coding problem has a considerable advantage compared to the channel coding problem. The encoder knows the information seen by the decoder (unlike channel coding there is no noise involved here). Therefore, the encoder can also decode and check

whether the decoding is successful or not. In case of failure, the encoder can retry the compression procedure by using a permutation of the source vector. This permutation is fixed a priori and is known both to the encoder as well as the decoder. In order to completely specify the system, the encoder must inform the decoder which permutation was finally used. This results in a small loss of rate but it brings down the probability of decoding failure. Note that the extra number of bits that need to be transmitted grows only logarithmically with the number of permutations used, but that the error probability decays exponentially as long as the various permuted source vectors look like independent source samples. With this trick one can make the curves essentially arbitrarily steep with a very small loss in rate (see Figure 5).

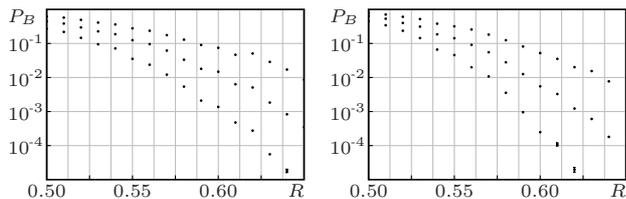


Fig. 5. Comparison of SC decoding for a Ber(0.11) source with 0, 1, and 2 bits for permutations. The performance curves are shown for $n = 10$ (left) and 11 (right). By increasing the number of permutations the curves can be made steeper and steeper.

2) *Slepian-Wolf*: Consider two $\text{Ber}(\frac{1}{2})$ sources X and Y . Assume that they are correlated as $\bar{X} = Y \oplus Z$, where $Z \sim \text{Ber}(p)$. Recall that the Slepian-Wolf rate region is the unbounded polytope described by $R_X > H(X) = 1$, $R_Y > H(Y) = 1$, $R_X + R_Y > H(X, Y) = 1 + h_2(p)$. The points $(R_X, R_Y) = (1, h_2(p))$ and $(R_X, R_Y) = (h_2(p), 1)$ are the so-called *corner points*. Because of symmetry it suffices to show how to achieve one such corner point (say $(1, h_2(p))$).

Let \bar{x} and \bar{y} denote N i.i.d. realizations of the two sources. The scheme using linear codes is the following: The encoder for X transmits \bar{x} as it is (since $H(X) = 1$, and so no compression is necessary). Let \mathbf{H} denote the parity-check matrix of a code designed for communication over the BSC(p). The encoder for Y computes the syndrome $\bar{s} = \mathbf{H}\bar{y}$ and transmits it to the receiver. At the receiver we know $\bar{s} = \mathbf{H}\bar{y}$ and \bar{x} , therefore we can compute $\bar{s}' = \mathbf{H}(\bar{x} \oplus \bar{y}) = \mathbf{H}\bar{z}$. The resulting problem of estimating \bar{z} is equivalent to the lossless compression of a $\text{Ber}(p)$ discussed in the previous section. Therefore, once again polar codes provide an efficient solution. The error probability curves under SC decoding are equal to the curves shown in the Figure 5 with 0 bits for permutations.

B. Lossy Source Coding

1) *Single User*: We do not know of a mapping that converts the lossy source coding problem to a channel coding problem. However, for the binary erasure source considered in [8], it was shown how to construct a “good” source code from a “good” (over the BEC) channel code. We briefly describe their construction here and show that polar codes achieve the optimal rate for zero distortion.

The source is a sequence of i.i.d. realizations of a random variable S taking values in $\{0, 1, *\}$ with $\Pr(S = 0) = \Pr(S = 1) = \frac{1}{2}(1 - \epsilon)$, $\Pr(S = *) = \epsilon$. The reconstruction alphabet is $\{0, 1\}$ and the distortion function is given by $d(*, 0) = d(*, 1) = 0$, $d(0, 1) = 1$. For zero distortion, the rate of the rate-distortion function is given by $R(D = 0) = 1 - \epsilon$. In [8] it was shown that the dual of a sequence of channel codes which achieve the capacity of the $\text{BEC}(1 - \epsilon)$ under BP decoding, achieve the rate-distortion pair for zero distortion using a message passing algorithm which they refer to as the erasure quantization algorithm. Polar codes achieve capacity under SC decoding. For communication over BEC, the performance under BP is at least as good as SC. Therefore, the dual of the polar codes designed for $\text{BEC}(1 - \epsilon)$ achieve the optimum rate for zero distortion using the erasure quantization algorithm. Here, we show that the dual polar codes achieve the optimum rate for zero distortion even under a suitably defined SC decoding algorithm.

The dual of a polar code is obtained by reversing the roles of the check and variable nodes of the trellis in Figure 2(a) and reversing the roles of the frozen and free bits. It is easy to see that $G_2^{\otimes n} G_2^{\otimes n} = \mathbf{I}$. This implies that the dual of a polar code is also a polar code. The *suitably* defined algorithm is given by SC decoding in the order $\pi(N - 1), \dots, \pi(0)$, opposite of the original decoding order. We refer to this as the dual order.

In [1], the probability of erasure for bit u_i under SC decoding is given by $Z_n^{(i)}(1 - \epsilon)$, computed as follows. Let the n -bit binary expansion of i be given by b_{n-1}, \dots, b_0 . Let $Z_0 = 1 - \epsilon$. The sequence $Z_1, \dots, Z_n = Z_n^{(i)}(1 - \epsilon)$ is recursively defined as follows:

$$Z_k = \begin{cases} Z_{k-1}^2, & \text{if } b_{k-1} = 1, \\ 1 - (1 - Z_{k-1})^2, & \text{if } b_{k-1} = 0. \end{cases} \quad (1)$$

From [3] we know that for any $R < \epsilon$, there exists an n_0 such that for $n \geq n_0$ we can find a set $I \subset \{0, \dots, 2^n - 1\}$ of size $2^n R$ satisfying $\sum_{i \in I} Z_n^{(i)}(1 - \epsilon) \leq 2^{-N^\beta}$ for any $\beta < \frac{1}{2}$. The set I is used as information bits. The complement of I denoted by F , is the set of frozen bits.

Let $\hat{Z}_n^{(i)}(\epsilon)$ denote the probability of erasure for bit u_i of the dual code used for $\text{BEC}(\epsilon)$. One can check that for the dual code with the dual order, the bit u_i is equivalent to the bit u_{N-i} of the original code with the original decoding order. Let $\hat{Z}_0 = \epsilon$. The recursive computation for $\hat{Z}_n^{(i)}$ is given by

$$\hat{Z}_k = \begin{cases} \hat{Z}_{k-1}^2, & \text{if } b_{k-1} = 0, \\ 1 - (1 - \hat{Z}_{k-1})^2, & \text{if } b_{k-1} = 1. \end{cases} \quad (2)$$

Lemma 7 (Duality for BEC): $Z_n^{(i)}(1 - \epsilon) + \hat{Z}_n^{(i)}(\epsilon) = 1$.

The proof follows through induction on n and using the equations (1) and (2). For the dual code the set I (information set for $\text{BEC}(1 - \epsilon)$) is used as frozen bits and the set F is used as information bits. Let \bar{x} be a sequence 2^n source realizations. The source vector \bar{x} needs to be mapped to a vector \bar{u}_F (information bits) such that $d(\bar{u}_F G_2^{\otimes n}, \bar{x}) = 0$. The following lemma shows that such a vector \bar{u}_F can be found using SC decoder with vanishing error probability.

Lemma 8: The probability of encoding failure for erasure quantization of the source S using the dual of the polar code designed for the BEC($1-\epsilon$) and SC decoding with dual order is bounded as $P_e(N) = o(2^{-N^\beta})$ for any $\beta < \frac{1}{2}$.

Proof: Let I and F be as defined above. The bits belonging to I are already fixed to 0 whereas the bits belonging to F are free to be set. Therefore an error can only occur if one of the bits belonging to I are set to a wrong value. However, if the SC decoding results in an erasure for these bits, these bits are also free to be set any value and this results in no error. Therefore the probability of error can be upper bounded by the probability that at least one of the bits in I is not an erasure which in turn can be upper bounded by $\sum_{i \in I} (1 - \hat{Z}_n^{(i)}(\epsilon)) = \sum_{i \in I} Z_n^{(i)}(1-\epsilon) = o(2^{-N^\beta})$, where the last equality follows from Theorem 2. ■

The fact that the dual code can be successfully applied to the erasure source, suggests to extend this construction to more general sources. Let us try a similar construction to encode the $\text{Ber}(\frac{1}{2})$ source with the Hamming distortion. To design a source code for distortion D we first design a rate $1 - h_2(p)$ polar code for the $\text{BSC}(p)$ where $p = h_2^{-1}(1 - h_2(D))$. The design consists of choosing the generator vectors from the rows of $G_2^{\otimes n}$ as explained in [1]. The source code is then defined by the corresponding dual code with the dual decoding order. Since the rate of the original code is $1 - h_2(p)$, the rate of the dual code is $h_2(p) = 1 - h_2(D)$. Figure 6 shows the rate-distortion performance of these codes for various lengths. As the lengths increase, empirically we observe that the performance approaches the rate-distortion curve. It is

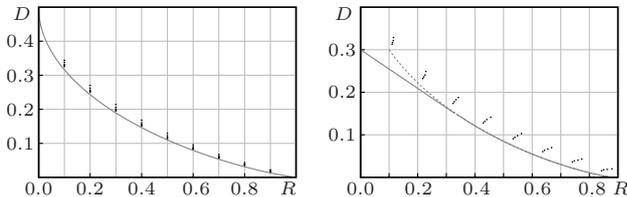


Fig. 6. (left) Rate-Distortion curves for a $\text{Ber}(0.11)$ source. The performance curves are shown for $n = 9, 11, 13, 15, 17$. (right) The dashed line is the curve $R_{WZ}(D)$. The solid line is the lower convex envelope of $R_{WZ}(D)$ and $(0, 0.3)$ for a $\text{Ber}(\frac{1}{2})$ source with receiver having a side information obtained through a $\text{BSC}(0.3)$. The performance curves are shown for $n = 9, 11, 13, 15$.

an interesting and challenging open problem to prove this observation rigorously.

2) *Wyner-Ziv:* Let X denote a $\text{Ber}(\frac{1}{2})$ source which we want to compress. The source is reconstructed at a receiver which has access to a side information Y correlated to X as $Y = X \oplus Z$ with $Z \sim \text{Ber}(p)$. Let \bar{x} and \bar{y} denote a sequence of N i.i.d. realizations of X and Y . Wyner and Ziv have shown that the rate distortion curve is given by the lower convex envelope of the curve $R_{WZ}(D) = h_2(D * p) - h_2(D)$ and the point $(0, p)$, where $D * p = D(1 - p) + (1 - D)p$. As discussed in [9], nested linear codes are required to tackle this problem. The idea is to partition the codewords of a code C_1 into cosets of another code C_2 . The code C_1 must be a good

source code and the code C_2 must be a good channel code.

Using polar codes, we can create this nested structure as follows. Let C_1 be a source code for distortion D (Bernoulli source, Hamming distortion) as described in the previous section. Let F_s be the set of frozen (fixed to 0) bits for this source code. Using this code we quantize the source vector \bar{x} . Let the resulting vector be \hat{x} . Note that the vector \hat{x} is given by $\hat{u}G_2^{\otimes n}$, where $\hat{u} = (\hat{u}_0, \dots, \hat{u}_{N-1})$ is such that $\hat{u}_i = 0$ for $i \in F_s$ and for $i \in F_s^c$, \hat{u}_i is defined by the source quantization.

For a sequence of codes achieving the rate-distortion bound, it was shown in [10] that the “noise” $\bar{x} \oplus \hat{x}$ added due to the quantization is comparable to a Bernoulli noise $\text{Ber}(D)$. Therefore, for all practical purposes, the side information at the receiver \bar{y} is statistically equivalent to the output of \hat{x} transmitted through a $\text{BSC}(D * p)$. Let F_c be the set of frozen bits of a channel code for the $\text{BSC}(D * p)$. Let the encoder transmit the bits $\hat{u}_{F_c \setminus F_s}$ to the receiver. Since the bits belonging to F_s (\hat{u}_{F_s}) are fixed to zero, the receiver now has access to \hat{u}_{F_c} and \bar{y} . The definition of F_c implies that the receiver is now able to decode \hat{u}_{F_c} (and hence \hat{x}) with vanishing error probability. To see the nested structure, note that the code C_2 and its cosets are the different channel codes defined by different values of $\bar{u}_{F_c \setminus F_s}$ and that these codes partition the source code C_1 . The capacity achieving property of polar codes implies that for N sufficiently large, $|F_c| < N(h_2(D * p) + \delta)$ for any $\delta > 0$. In addition, if we assume that polar codes achieve the rate-distortion bound as conjectured in the previous section and $F_s \subseteq F_c$, then for N sufficiently large, the rate required to achieve a distortion D is $\frac{1}{N}|F_c \setminus F_s| \leq h_2(D * p) - h_2(D) + \delta$ for any $\delta > 0$. This would show that polar codes can be used to efficiently realize the Wyner-Ziv scheme. The performance of the above scheme for various lengths is shown in Figure 6.

REFERENCES

- [1] E. Arkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *submitted to IEEE Trans. Inform. Theory*, 2008.
- [2] G. D. Forney, Jr., “Codes on graphs: Normal realizations,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.
- [3] E. Arkan and E. Telatar, “On the rate of channel polarization,” July 2008, available from “<http://arxiv.org/pdf/0807.3917>”.
- [4] E. Arkan, “A performance comparison of Polar codes and Reed-Muller codes,” *IEEE Communications Letters*, vol. 12, no. 6, 2008.
- [5] M. Schwartz and A. Vardy, “On the stopping distance and the stopping redundancy of codes,” *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 922–932, Mar. 2006.
- [6] P. E. Allard and E. W. Bridgewater, “A source encoding technique using algebraic codes,” June 1972, pp. 201–203.
- [7] E. Weiss, “Compression and coding,” *IRE Trans. Inform. Theory*, pp. 256–257, 1962.
- [8] E. Martinian and J. Yedidia, “Iterative quantization using codes on graphs,” in *Proc. of the Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, USA, 2003.
- [9] R. Zamir, S. Shamai, and U. Erez, “Nested linear/lattice codes for structured multiterminal binning,” *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1216, 2002.
- [10] U. Erez and R. Zamir, “Bounds on the ϵ -covering radius of linear codes with applications to self noise in nested Wyner-Ziv coding,” Feb. 2002, dept. Elec. Eng-Syst., Tel-Aviv Univ., Tech. Rep.