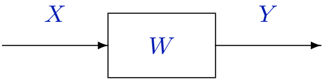
1. Polar Codes

Gadiel Seroussi

October 26, 2022

- Comparison with our thought experiment?
- Polar encoder
- Channel ordering on the BEC
- Encoding example: $N = 8,\ K = 4$ on BEC(0.5)
- Successive cancellation decoding (SCD)
- Performance of SCD
- SCD issues
- A reflection on polar codes and Shannon's paradise
- Polar codes: development

Discrete binary-input channels

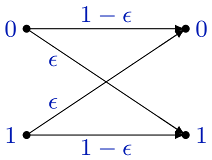


- Input alphabet: $\mathcal{X} = \{0, 1\}$.
- Output alphabet: $\mathcal{Y}$.
- Transition probabilities: $W(y|x), \;\; y \in \mathcal{Y}, \; x \in \mathcal{X}$.
- Capacity: $C(W) = \sup_{P_X} I(X;Y)$.
- Throughout this unit, we assume all raw channels are *memoryless*.

Symmetric binary-input channels

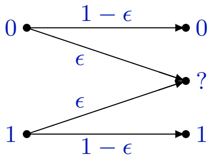
A binary-input discrete memoryless channel $W : \{0,1\} \to \mathcal{Y}$ is *symmetric* if there exists a permutation $\pi : \mathcal{Y} \to \mathcal{Y}$ such thatt $\pi = \pi^{-1}$ and $W(y|x) = W(\pi(y)|\bar{x})$ for all $y \in \mathcal{Y}$.

Binary symmetric channel
BSC(ϵ)



Capacity $C(W) = 1 - H(\epsilon)$

Binary erasure channel
BEC(ϵ)



Capacity $C(W) = 1 - \epsilon$

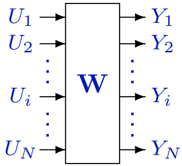
For symmetric channels, capacity is achieved with a uniform input distribution:

$$C(W) = I(X;Y) \text{ with } X \sim \text{ Bernoulli}(1/2) \;;$$

all logs are base-2 $\implies$ $0 \le C(W) \le 1$.

A guessing game

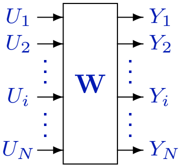
Say we want to guess (decode) the value of a binary vector U^N after observing a related random vector Y^N.



- For example, U^N may be a random codeword from a code, and Y^N a channel's output when the input is U^N.

A guessing game

Say we want to guess (decode) the value of a binary vector U^N after observing a related random vector Y^N.



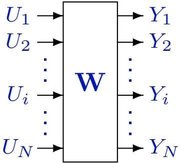
- For example, U^N may be a random codeword from a code, and Y^N a channel's output when the input is U^N.
- To minimize the probability of decoding error, upon observing y^N,we choose the value u^n that *maximizes*

$$p(u^N|y^N) = \prod_{i=1}^{N} p(u_i|y^n, u^{i-1}).$$

prob. law governing relation between U^N and Y^N; shortcut for $P_{U^N,Y^N}(\cdot)$.

A guessing game

Say we want to guess (decode) the value of a binary vector U^N after observing a related random vector Y^N.



- For example, U^N may be a random codeword from a code, and Y^N a channel's output when the input is U^N.
- To minimize the probability of decoding error, upon observing y^N,we choose the value u^n that *maximizes*

$$p(u^N|y^N) = \prod_{i=1}^{N} p(u_i|y^n, u^{i-1}).$$

prob. law governing relation between U^N and Y^N; shortcut for $P_{U^N,Y^N}(\cdot)$.

- Channel interpretation: define *bit channels*

$$W_i : U_i \to (Y^N, U^{i-1})\,.$$

Then,

$$C(\mathbf{W}) = I(U^N; Y^N) = \sum_{i=1}^{N} I(U_i; Y^N, U^{i-1}) = \sum_{i=1}^{N} C(W_i)\,.$$

Extremal channels

$$p(u^N|y^N) = \prod_{i=1}^{N} p(u_i|y^n, u^{i-1}).$$

Two extreme cases for $p(u_i|y^n, u^{i-1})$:

1. $p(u_i|y^n, u^{i-1}) = \mathbf{1}_{u_i}$: u_i is a *function* $F(y^n, u^{i-1})$;
   perfect channel, capacity $C(W_i) = 1$.
2. $p(u_i|y^n, u^{i-1}) = \frac{1}{2}$: y^n, u^{i-1} *provide no information* on u_i;
   useless channel, capacity $C(W_i) = 0$.

A thought experiment

Assume $p(\cdot|\cdot,\cdot)$ is such that all channels W_i are either *perfect* or *useless*. Let $[N] = \{1, 2, \ldots, N\}$, and

$$\mathcal{A} = \{i \in [N] \mid W_i \text{ is perfect}\}; \;\; \mathcal{A}^c = [N] \setminus \mathcal{A}\,.$$

Assume also that if $i \in \mathcal{A}^c$, a *genie* provides us with the values u_i. Then, we can decode u^n perfectly from y^n with the following sequential algorithm.

For $i = 1, 2, \ldots, N$:

- If $i \in \mathcal{A}^c$, get u_i from the genie.
- If $i \in \mathcal{A}$, compute $u_i = F(y^n, u^{i-1})$.

Of course, this situation is unrealistic in practice (both the nature of the channels, and the genie).

Just a thought experiment?

Of course, the thought experiment is unrealistic in practice. However, taking advantage of a *channel polarization* phenomenon, with *polar codes* one can

Just a thought experiment?

Of course, the thought experiment is unrealistic in practice. However, taking advantage of a *channel polarization* phenomenon, with *polar codes* one can

- start with N independent, identical, imperfect (but not useless) binary-input symmetric channels,
- apply a transformation into N inter-dependent channels that are, asymptotically (as $N \to \infty$), partitioned as $\mathcal{A} \cup \mathcal{A}^c$ above, with channels in $\mathcal{A}$ being arbitrarily close to perfect, and $|\mathcal{A}|/N \to C(W)$,
- find an appropriate "genie" for free,
- encode with a code rate attaining, asymptotically, the capacity of the original channels, and with probability of decoding error $O(2^{-\sqrt{N}})$,
- provide efficient encoding and decoding algorithms (complexity $O(N \log N)$),
- use *deterministic* constructions throughout, with their properties mathematically proven.

This set of features was *unprecedented* when the work was first published.

References

The seminal paper:

- Erdal Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", *IEEE Trans. Info. Theory*, **55**, pp. 3051–3073, July 2009.

Significant improvements soon after:

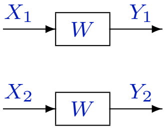
- R Mori, T. Tanaka, "Performance of polar codes with the construction using density evolution", *IEEE Communication Letters*, **13**(7), pp. 519–521, 2009.
- I. Tal, A. Vardi, "How to Construct Polar Codes", *IEEE Trans. on Info. Theory*, **59**, pp. 6562–6582, Oct. 2013.
- Many others ...

A good tutorial:

- Eren Şaşoğlu, "Polarization and Polar Codes", *Foundations and Trends in Communications and Information Theory*, **8**(4), pp. 259–381, Oct. 2012.

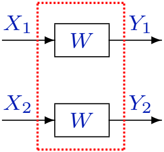
A basic transform

Begin with two independent copies (or uses) of W



A basic transform

Begin with two independent copies (or uses) of W

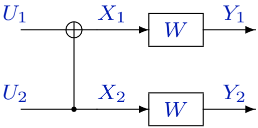


$W^2 : (X_1, X_2) \to (Y_1, Y_2)$

$C(W^2) = 2C(W)$

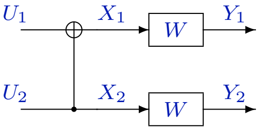
A basic transform

Combine the channels



A basic transform

Combine the channels



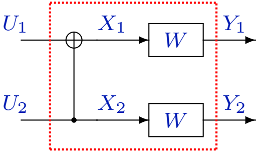
We have

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = G_2 \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}, \text{ with } G_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \ (G_2 = G_2^{-1})$$

(arithmetic $\bmod 2$).

A basic transform

Combine the channels



$$W_{\text{vec}} : (U_1, U_2) \to (Y_1, Y_2)$$

We have

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = G_2 \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}, \text{ with } G_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \ (G_2 = G_2^{-1})$$

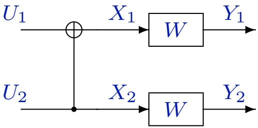
(arithmetic $\mod 2$).

- Because of the created dependencies, the *vector channel* $W_{\text{vec}} : (U_1, U_2) \to (Y_1, Y_2)$ *cannot* be interpreted as two independent uses of a single channel, as $W^2 : (X_1, X_2) \to (Y_1, Y_2)$ can.
- Since G_2 is invertible, we have

$$C(W_{\text{vec}}) = C(W^2) = 2C(W)\,.$$

Decomposition into bit channels

Combined channels



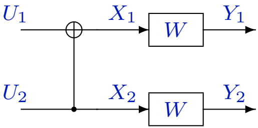
We have

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = G_2 \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}, \text{ with } G_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad (G_2 = G_2^{-1})$$

(arithmetic $\bmod 2$).

Decomposition into bit channels

Combined channels



We have

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = G_2 \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}, \text{ with } G_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad (G_2 = G_2^{-1})$$

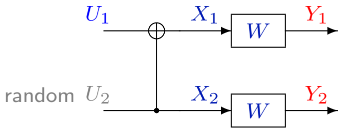
(arithmetic $\bmod 2$).

Define the *bit channels*

$$W_1 : U_1 \to (Y_1, Y_2)$$
$$W_2 : U_2 \to (Y_1, Y_2, U_1)$$

First bit channel

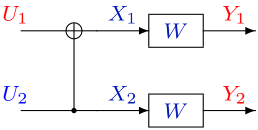
$W_1 : U_1 \to (Y_1, Y_2)$



$$C(W_1) = I(U_1; Y_1, Y_2) = H(U_1) - H(U_1|Y_1, Y_2)$$

Second bit channel

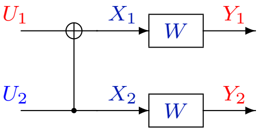
$W_2 : U_2 \to (Y_1, Y_2, U_1)$



$$C(W_2) = I(U_2; Y_1, Y_2, U_1) = H(U_2) - H(U_2|Y_1, Y_2, U_1)$$
$$= 1 - H(U_2|Y_1, Y_2, U_1) \geq 1 - H(U_2|Y_2) = 1 - H(X_2|Y_2) = C(W).$$

Second bit channel

$W_2 : U_2 \to (Y_1, Y_2, U_1)$



$$C(W_2) = I(U_2; Y_1, Y_2, U_1) = H(U_2) - H(U_2|Y_1, Y_2, U_1)$$
$$= 1 - H(U_2|Y_1, Y_2, U_1) \geq 1 - H(U_2|Y_2) = 1 - H(X_2|Y_2) = C(W).$$

Recall

$$C(W_1) + C(W_2) = C(W_{\mathsf{vec}}) = 2C(W) \,.$$

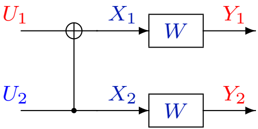
Therefore,

$$C(W_1) \leq C(W) \leq C(W_2).$$

In fact, inequalities are *strict* if $0 < C(W) < 1$.

Second bit channel

$W_2 : U_2 \to (Y_1, Y_2, U_1)$



$$C(W_2) = I(U_2; Y_1, Y_2, U_1) = H(U_2) - H(U_2|Y_1, Y_2, U_1)$$
$$= 1 - H(U_2|Y_1, Y_2, U_1) \geq 1 - H(U_2|Y_2) = 1 - H(X_2|Y_2) = C(W).$$

Recall

$$C(W_1) + C(W_2) = C(W_{\text{vec}}) = 2C(W) .$$

Therefore,

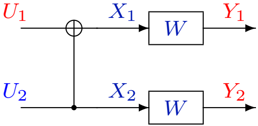
$$C(W_1) \leq C(W) \leq C(W_2).$$

In fact, inequalities are *strict* if $0 < C(W) < 1$.

The transformation took two identical channels, and transformed them into a pair where one is better than the original, and one is worse.

Second bit channel

$W_2 : U_2 \to (Y_1, Y_2, U_1)$



$$C(W_2) = I(U_2; Y_1, Y_2, U_1) = H(U_2) - H(U_2|Y_1, Y_2, U_1)$$
$$= 1 - H(U_2|Y_1, Y_2, U_1) \geq 1 - H(U_2|Y_2) = 1 - H(X_2|Y_2) = C(W).$$

Recall

$$C(W_1) + C(W_2) = C(W_{\text{vec}}) = 2C(W)\,.$$

Therefore,

$$C(W_1) \leq C(W) \leq C(W_2).$$

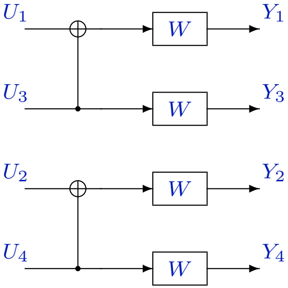
In fact, inequalities are *strict* if $0 < C(W) < 1$.

The transformation took two identical channels, and transformed them into a pair where one is better than the original, and one is worse.

Rename: $W^- \triangleq W_1, \quad W^+ \triangleq W_2$.

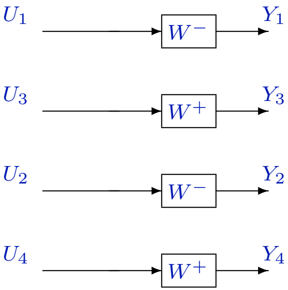
Double down on the construction

First, duplicate the basic transform



Double down on the construction

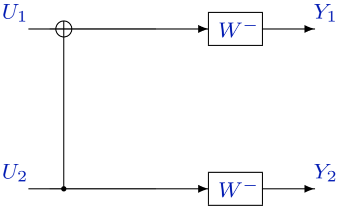
Reinterpret



A pair of W^- and a pair of W^+.

Double down on the construction

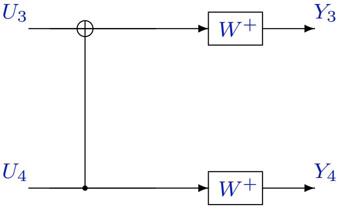
Apply the basic transform to each pair



Get W^{--}, W^{-+}

Double down on the construction

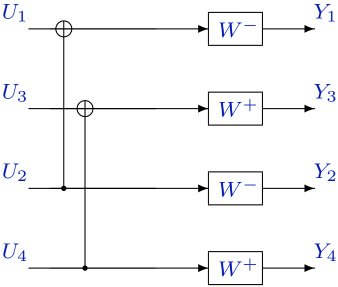
Apply the basic transform to each pair



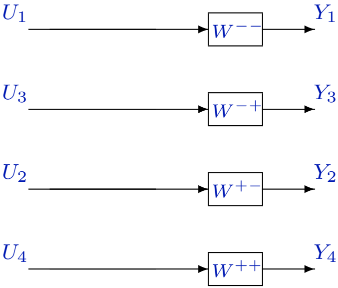
Get W^{+-}, W^{++}

Double down on the construction

Apply the basic transform to each pair



Double down on the construction



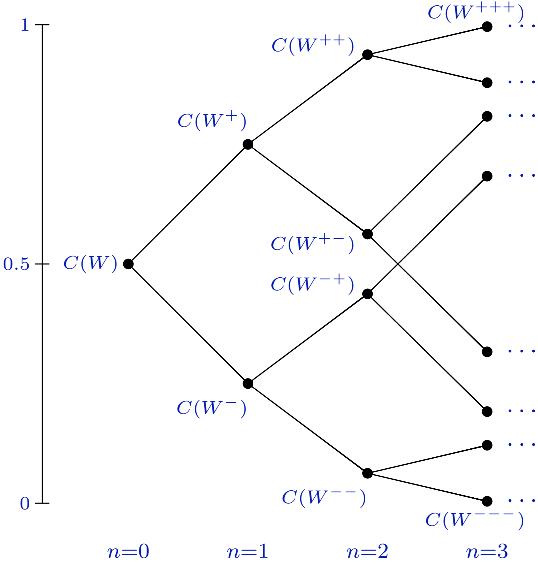
As for capacities, we know that in nontrivial cases, we have (abusing notation):

$$W^{--} < W^{-} < W < W^{+} < W^{++}$$

and also

$$W^{--} < W^{-} < W^{-+}, \quad W^{+-} < W^{+} < W^{++}$$

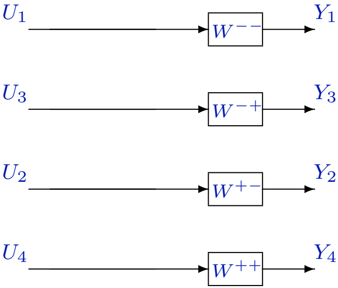
Recursive application of the transform



Recursive application of the transform continues to pull channels apart: *bad channels become worse, good channels become better*.

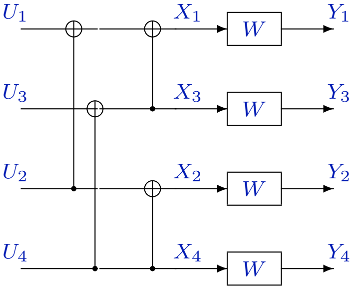
This process is called *polarization*.

Back to the $N = 4$ case



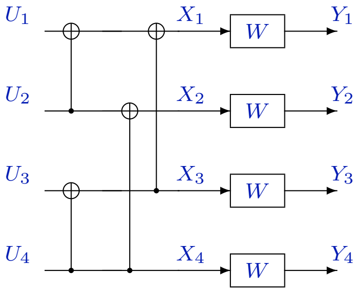
Back to the $N = 4$ case

Opening the boxes, detailed $N = 4$ structure



Back to the $N = 4$ case

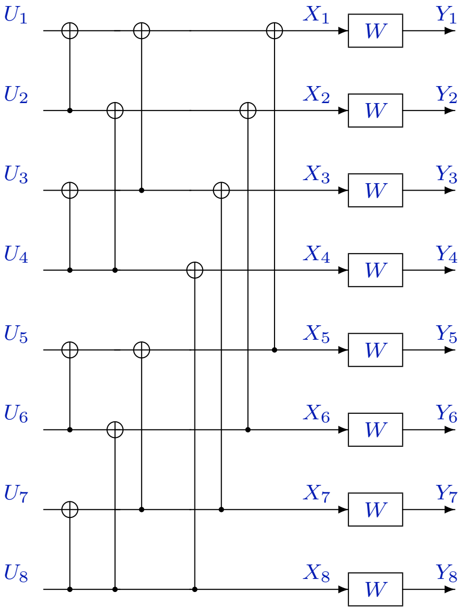
Put variables in standard order



We continue this construction recursively:
take two structures of length $N = 2^n$, construct one of length $2N = 2^{n+1}$.

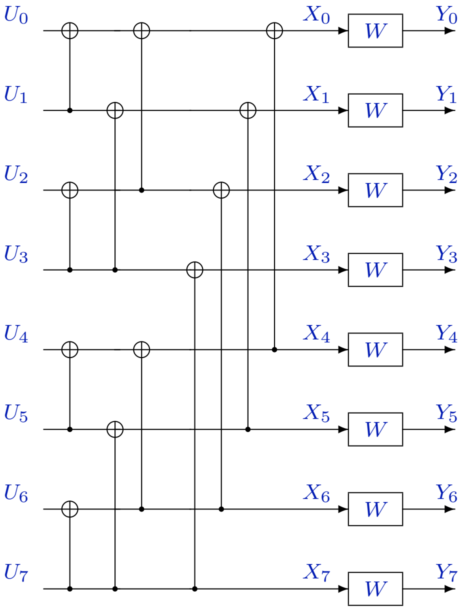
For each bit channel W^s in the length–N construction, where $s \in \{-,+\}^n$, we create channels W^{s-} and W^{s+} in the length–$2N$ construction.

Case $N = 8$

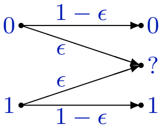


U1
U2
U3
U4
U5
U6
U7
U8
X1
X2
X3
X4
X5
X6
X7
X8
W
W
W
W
W
W
W
W
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8

Case $N = 8$ from 0



Binary erasure channels



Capacity $C(W) = 1 - \epsilon$.

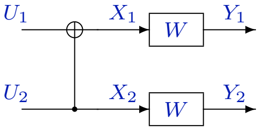
Generalization: A binary-input symmetric channel $W^* = \{0,1\} \to \mathcal{Y}$ is a *binary erasure channel* iff for each $y \in \mathcal{Y}$, either $W^*(y|0)W^*(y|1) = 0$ or $W^*(y|0) = W^*(y|1)$. In the latter case, y is called an *erasure symbol*.

Clearly, the usual BEC satisfies the definition. And so do the bit channels W^- and W^+, as we check next.

Binary erasure channels

For each $y \in \mathcal{Y}$, either $W^*(y|0)W^*(y|1) = 0$ or $W^*(y|0) = W^*(y|1)$. In the latter case, y is called an *erasure symbol*.

Let $a, b, c, d \in \{0, 1\},\ \nu \in \{(b, ?), (?, b)\}$.



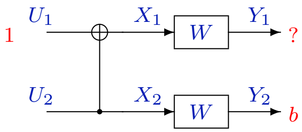
$W^- : U_1 \to (Y_1, Y_2)$.
$p(\nu|0) = p(\nu|1) = \epsilon(1-\epsilon)$,
$p(?, ?|a) = \epsilon^2,\ \ p(a, b|c) = (1-\epsilon)^2\, \mathbf{1}_{a=b+c}$.

$W^+ : U_2 \to (Y_1, Y_2, U_1)$.
$p(a, ?, b|c) = \epsilon(1-\epsilon)\, \mathbf{1}_{a=b+c}$,
$p(?, a, b|c) = \epsilon(1-\epsilon)\, \mathbf{1}_{a=c}$,
$p(?, ?, a|b) = \epsilon^2$,
$p(a, b, c|d) = (1-\epsilon)^2\, \mathbf{1}_{b=d \text{ and } a=c+d}$.

Binary erasure channels

For each $y \in \mathcal{Y}$, either $W^*(y|0)W^*(y|1) = 0$ or $W^*(y|0) = W^*(y|1)$. In the latter case, y is called an *erasure symbol*.

Let $a, b, c, d \in \{0, 1\},\ \nu \in \{(b, ?), (?, b)\}$.



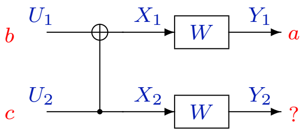
$W^- : U_1 \to (Y_1, Y_2)$.
$p(\nu|0) = p(\nu|1) = \epsilon(1-\epsilon)$, ←
$p(?, ?|a) = \epsilon^2,\ \ p(a, b|c) = (1-\epsilon)^2\, \mathbf{1}_{a=b+c}$.

$W^+ : U_2 \to (Y_1, Y_2, U_1)$.
$p(a, ?, b|c) = \epsilon(1-\epsilon)\, \mathbf{1}_{a=b+c}$,
$p(?, a, b|c) = \epsilon(1-\epsilon)\, \mathbf{1}_{a=c}$,
$p(?, ?, a|b) = \epsilon^2$,
$p(a, b, c|d) = (1-\epsilon)^2\, \mathbf{1}_{b=d \text{ and } a=c+d}$.

Binary erasure channels

For each $y \in \mathcal{Y}$, either $W^*(y|0)W^*(y|1) = 0$ or $W^*(y|0) = W^*(y|1)$. In the latter case, y is called an *erasure symbol*.

Let $a, b, c, d \in \{0, 1\},\ \nu \in \{(b, ?), (?, b)\}$.



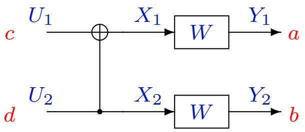
$W^- : U_1 \to (Y_1, Y_2)$.
$p(\nu|0) = p(\nu|1) = \epsilon(1 - \epsilon)$,
$p(?, ?|a) = \epsilon^2,\ \ p(a, b|c) = (1 - \epsilon)^2\, \mathbf{1}_{a=b+c}$.

$W^+ : U_2 \to (Y_1, Y_2, U_1)$.
$p(a, ?, b|c) = \epsilon(1 - \epsilon)\, \mathbf{1}_{a=b+c}$, ←
$p(?, a, b|c) = \epsilon(1 - \epsilon)\, \mathbf{1}_{a=c}$,
$p(?, ?, a|b) = \epsilon^2$,
$p(a, b, c|d) = (1 - \epsilon)^2\, \mathbf{1}_{b=d \text{ and } a=c+d}$.

Binary erasure channels

For each $y \in \mathcal{Y}$, either $W^*(y|0)W^*(y|1) = 0$ or $W^*(y|0) = W^*(y|1)$. In the latter case, y is called an *erasure symbol*.

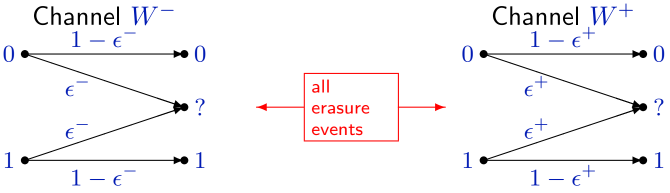
Let $a, b, c, d \in \{0, 1\}, \ \nu \in \{(b, ?), (?, b)\}$.



$W^- : U_1 \to (Y_1, Y_2)$.
$p(\nu|0) = p(\nu|1) = \epsilon(1-\epsilon)$,
$p(?, ?|a) = \epsilon^2, \ \ p(a, b|c) = (1-\epsilon)^2 \, \mathbf{1}_{a=b+c}$.

$W^+ : U_2 \to (Y_1, Y_2, U_1)$.
$p(a, ?, b|c) = \epsilon(1-\epsilon) \, \mathbf{1}_{a=b+c}$,
$p(?, a, b|c) = \epsilon(1-\epsilon) \, \mathbf{1}_{a=c}$,
$p(?, ?, a|b) = \epsilon^2$,
$p(a, b, c|d) = (1-\epsilon)^2 \, \mathbf{1}_{b=d \text{ and } a=c+d}$. ←

Binary erasure channels



$\epsilon^- = 2\epsilon - \epsilon^2$ $\qquad$ $\epsilon^+ = \epsilon^2$

Verification:

$$\epsilon^+ - \epsilon = \epsilon^2 - \epsilon < 0; \quad \epsilon^- - \epsilon = \epsilon - \epsilon^2 > 0,$$

so,

$$\epsilon^+ < \epsilon < \epsilon^- \quad \Longrightarrow \quad C(W^+) > C(W) > C(W^-)$$

This gives an obvious recursion to compute ϵ^s for $s \in \{-,+\}^n$, the erasure probability of W^s, from which $C(W^s)$ follows.

Binary erasure channels

Example. $\epsilon = 0.5, \;\; C(W) = 0.5$.

$n = 1 : C(W^{-}) = 0.25,\, C(W^{+}) = 0.75,$

$n = 2 : C(W^{--}) = 0.0625,\, C(W^{-+}) = 0.4375,\, C(W^{+-}) = 0.5625,\, C(W^{++}) = 0.9375,$

$n = 3 : 0.00390625,\; 0.12109375,\; 0.19140625,\; 0.68359375,$

$0.31640625,\; 0.80859375,\; 0.87890625,\; 0.99609375$

Binary erasure channels

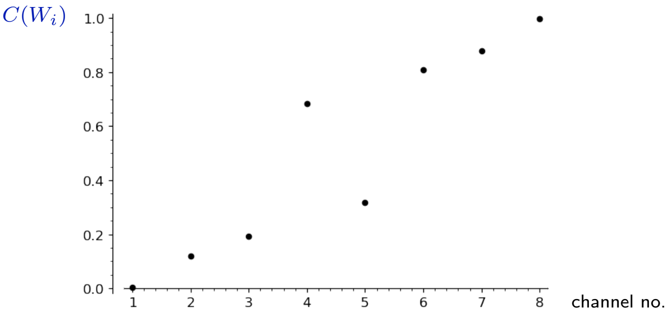
Example. $\epsilon = 0.5, \ \ C(W) = 0.5$.

$n = 1 : C(W^-) = 0.25, \, C(W^+) = 0.75,$

$n = 2 : C(W^{--}) = 0.0625, \, C(W^{-+}) = 0.4375, \, C(W^{+-}) = 0.5625, \, C(W^{++}) = 0.9375,$

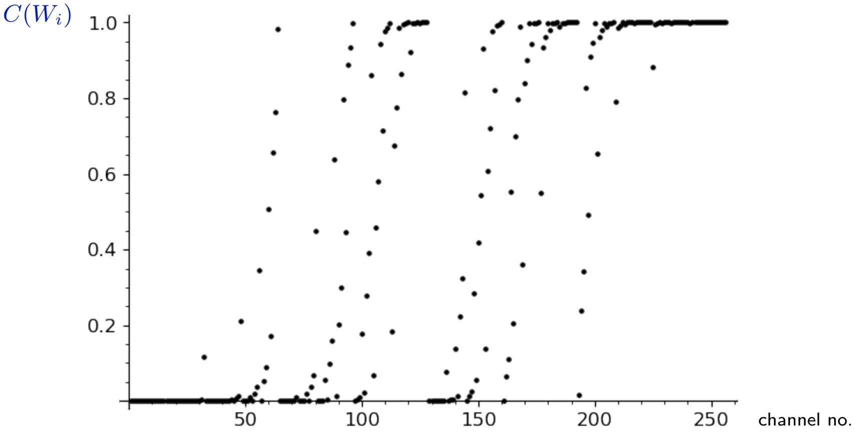
$n = 3 : 0.00390625, \ 0.12109375, \ 0.19140625, \ 0.68359375,$

$\quad 0.31640625, \ 0.80859375, \ 0.87890625, \ 0.99609375$



Capacity of bit channels for $n = 3, \ \epsilon = 0.5$

Binary erasure channels



Capacity of bit channels for $n = 8, \ \epsilon = 0.5$

Another relabeling

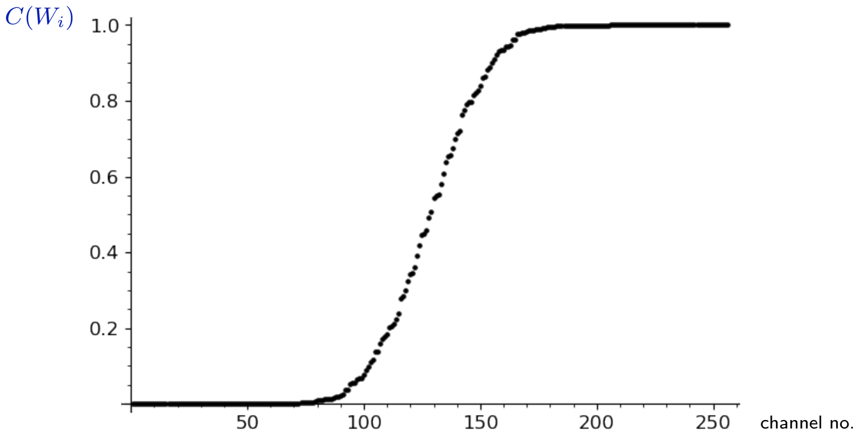
Let $s \in \{+,-\}^n$. Interpret '$+$' as a '1', '-' as a '0', and let i be the integer represented in binary by s. We will index the channels with i, and denote

$$W_i = W^s, \quad 0 \le i < 2^n \,.$$

Example, for $n = 3$:

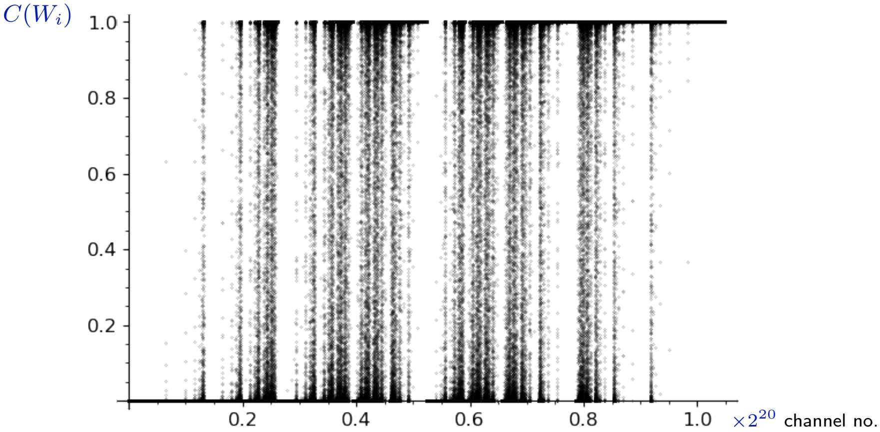
$$\begin{aligned} W_0 &= W^{---}, \\ W_1 &= W^{--+}, \\ \vdots \quad & \quad \vdots \\ W_7 &= W^{+++} \end{aligned}$$

Binary erasure channels



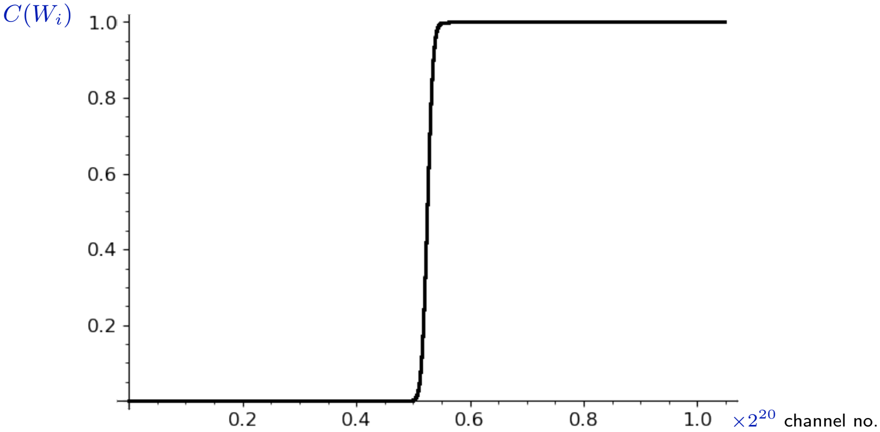
Capacity of bit channels for $n = 8,\ \epsilon = 0.5$ (sorted)

Binary erasure channels



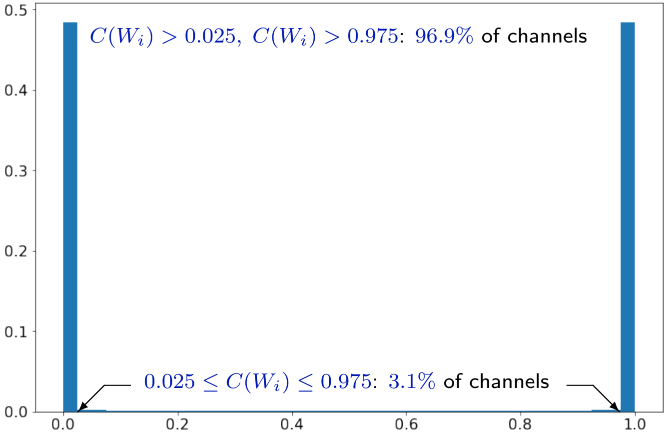
Capacity of bit channels for $n = 20,\ \epsilon = 0.5$

Binary erasure channels



Capacity of bit channels for $n = 20,\ \epsilon = 0.5$ (sorted)

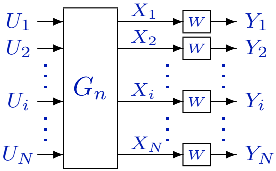
Binary erasure channels



Histogram of bit channel capacities for $n = 2,\ \epsilon = 0.5$ (bin size = 0.025)

Polarization

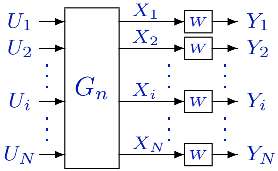
What we have done:



G_N is an invertible $N \times N$ linear transformation, $N = 2^n$. G_N polarizes the bit channels $W_i : U_i \to (Y^N, U^{i-1})$.

Polarization

What we have done:



G_N is an invertible $N \times N$ linear transformation, $N = 2^n$. G_N polarizes the bit channels $W_i : U_i \to (Y^N, U^{i-1})$.
Let $A \otimes A$ denote the *Kronecker product* of a matrix A with itself. E.g.,

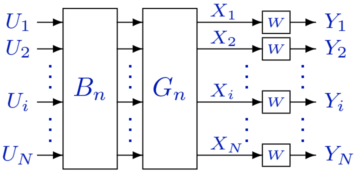
$$G_2 \otimes G_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes G_2 = \begin{bmatrix} G_2 & G_2 \\ \mathbf{0} & G_2 \end{bmatrix} = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Then,

$$G_N = G_2^{\otimes n} = \underbrace{G_2 \otimes G_2 \otimes \cdots \otimes G_2}_{n \text{ times}} .$$

Polarization–input permutation

A more general description:



where B_n is an $N \times N$ permutation matrix, and

$$G_N = G_2^{\otimes n} = \underbrace{G_2 \otimes G_2 \otimes \cdots \otimes G_2}_{n \text{ times}} .$$

Polarization

- We saw polarization at work for the BEC.
- In fact, G_N polarizes a *broad class of memoryless discrete channels*.
- Moreover, a *random* $N \times N$ transformation will, with high probability, achieve polarization. The advantage of G_N is in its recursive structure, enabling efficient construction and encoding/decoding algorithms.
- The following applies to G_N and binary-input symmetric channels:

Polarization theorems

Theorem (Polarization, Arikan 2007)

The bit-channel capacities $C(W_i)$ polarize: For any $\delta \in (0,1)$,

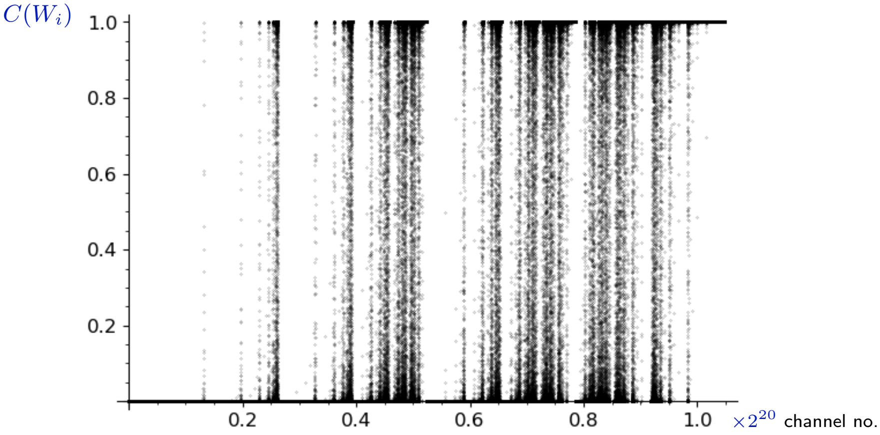
$$\frac{\textit{no. channels with } C(W_i) > 1-\delta}{N} \xrightarrow{N\to\infty} C(W)$$

$$\frac{\textit{no. channels with } C(W_i) < \delta}{N} \xrightarrow{N\to\infty} 1 - C(W)$$

Theorem (Rate of polarization, Arikan-Telatar 2008)

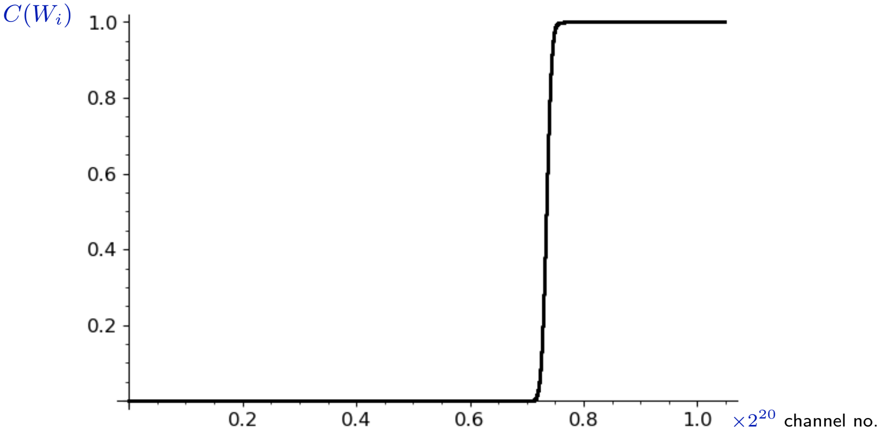
Above theorem holds with $\delta \approx 2^{-\sqrt{N}}$.

Polarization: more examples



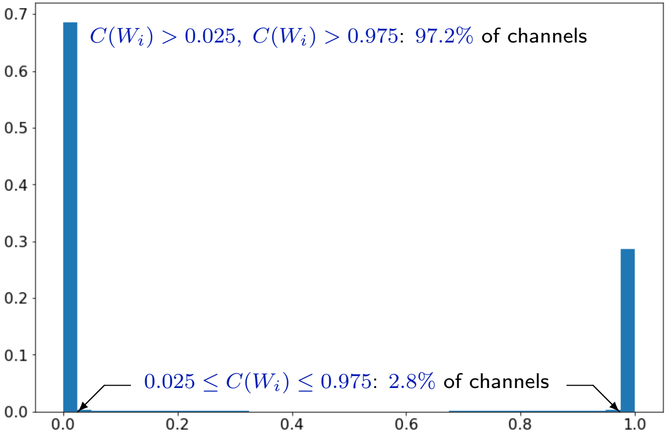
Capacity of bit channels for $n = 20$, $\epsilon = 0.7$

Polarization: more examples



Capacity of bit channels for $n = 20$, $\epsilon = 0.7$ (sorted)

Polarization: more examples



Histogram of bit channel capacities for $n = 20$, $\epsilon = 0.7$ (bin size $=$ 0.025)

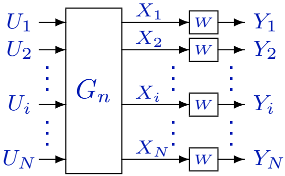
So, where are the codes?

We define a linear $[N, K]$ *polar code*, $K \leq N$, for a given raw channel W.

So, where are the codes?

We define a linear $[N, K]$ *polar code*, $K \leq N$, for a given raw channel W.

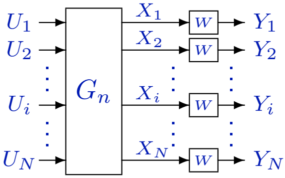
- Compute the capacities $C(W_i),\ 1 \leq i \leq N$.
- Find the set of K indices $\mathcal{A} = \{i_1, i_2, \ldots, i_K\}$ with the largest capacities.
  Let $\mathcal{A}^c = [N] \backslash \mathcal{A}$ (recall $[N] = \{1, 2, \ldots, N\}$).



So, where are the codes?

We define a linear $[N, K]$ *polar code*, $K \leq N$, for a given raw channel W.

- Compute the capacities $C(W_i),\ 1 \leq i \leq N$.
- Find the set of K indices $\mathcal{A} = \{i_1, i_2, \ldots, i_K\}$ with the largest capacities.
  Let $\mathcal{A}^c = [N] \backslash \mathcal{A}$ (recall $[N] = \{1, 2, \ldots, N\}$).



  - Use $U_{i_1}, U_{i_2}, \ldots U_{i_K}$ as *information symbols*.
  - Set the symbols $U_i,\ i \in \mathcal{A}^c$ to fixed binary values forming an $N{-}K$-vector $\mathbf{u}_{\mathcal{A}^c}$, *known to the decoder*. These are referred to as *frozen* bits.
  - The encoding is $U_{\mathcal{A}} \to X^N$. If $\mathbf{u}_{\mathcal{A}}$ is a message K-vector, then the corresponding codeword is

$$\mathbf{x} = \mathbf{u}_{\mathcal{A}} G_N^{\mathcal{A}} + \mathbf{u}_{\mathcal{A}^c} G_N^{\mathcal{A}^c} = \mathbf{u}_{\mathcal{A}} G_N^{\mathcal{A}} + \mathbf{v},$$

    where $G_N^{\mathcal{B}}$ consists of the rows of G_N with indices in $\mathcal{B},\ \mathcal{B} \subseteq [N]$. The vector $\mathbf{v}$ is fixed. If we set the frozen bits to zero, then $\mathbf{v} = 0$, and the code is *linear*. Otherwise, it is a *coset* of a linear code. This encoding is not necessarily systematic.

Comparison with our thought experiment?

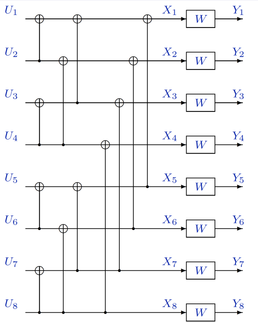
Notice the analogy to our thought experiment.

Thought experiment	Polar code
All channels in $\mathcal{A}$ are perfect.	All channels in $\mathcal{A}$ are perfect in the limit, or close to perfect for finite N, as long as $K/N < C(W)$.
All channels in $\mathcal{A}^c$ are useless.	$N\,C(W)$ channels in $\mathcal{A}^c$ are useless in the limit, but we freeze all of $\mathcal{A}^c$.
Bits corresponding to $\mathcal{A}^c$ are provided by a genie	Bits corresponding to $\mathcal{A}^c$ are fixed and known to the decoder (we are our own genie).
Decode u^N perfectly with a simple sequential procedure	Decode u^N with high probability with a relatively simple sequential procedure (*successive cancellation decoding—SCD*).

Polar encoder

- *Complexity.*
  - Straightforward computation is $O(N^2)$.
  - However, the recursive structure of the transform G_N allows for a fast $O(N \log N)$ computation.

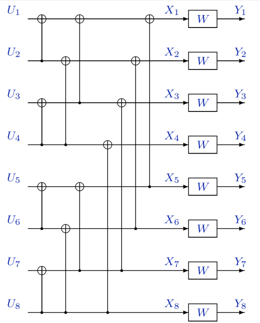
$$f(N) = 2f\left(\frac{N}{2}\right) + \frac{N}{2} = 4f\left(\frac{N}{4}\right) + 2\frac{N}{2}$$
$$= \cdots = Nf(1) + \frac{N}{2}\log N$$



Polar encoder

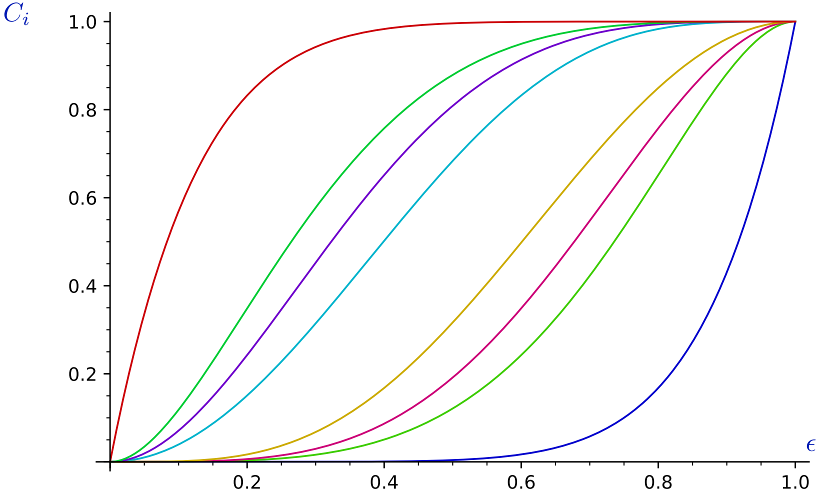
- *Complexity.*
  - Straightforward computation is $O(N^2)$.
  - However, the recursive structure of the transform G_N allows for a fast $O(N \log N)$ computation.

$$f(N) = 2f\left(\frac{N}{2}\right) + \frac{N}{2} = 4f\left(\frac{N}{4}\right) + 2\frac{N}{2}$$
$$= \cdots = Nf(1) + \frac{N}{2}\log N$$



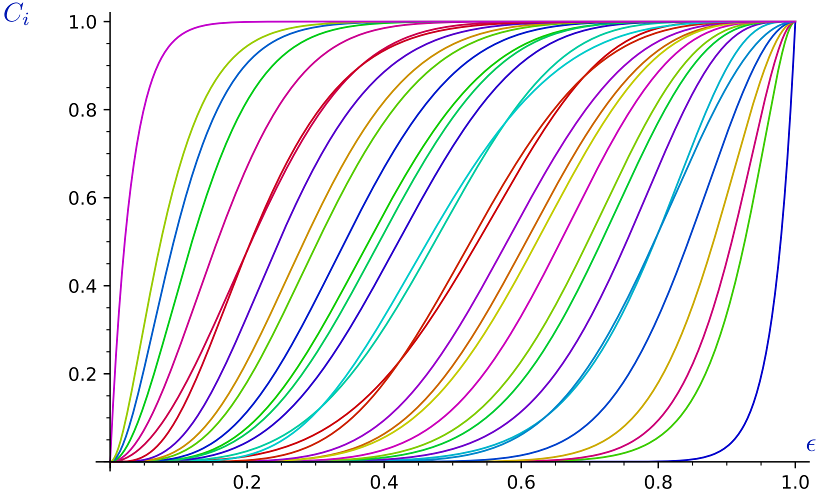
- *Bit selection:* selection of the information coordinates.
  - For a given code dimension $K = RN$, we need to select the K bit channels with highest capacities, given a known raw channel W.
  - Although easier for the BEC, since one can compute the capacity with an explicit recursion (with the parameter ϵ known), it is still a nontrivial problem if ϵ changes.
    - In principle, one would need to recompute the capacities and the channel ordering for each value of ϵ.
  - Even more complicated for other channels. However, good algorithms and approximations have been developed and work.

Channel ordering on the BEC



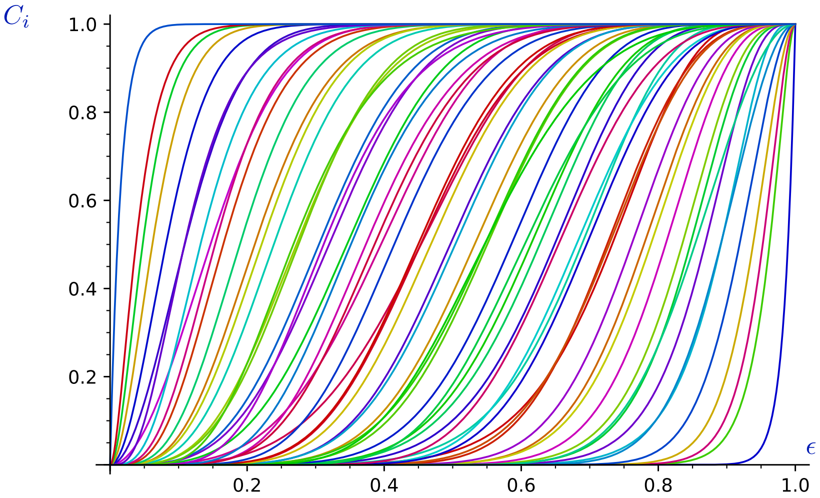
Capacities $C_i = C(W_i)$ of bit channels as a function of ϵ, $N = 8$

Channel ordering on the BEC



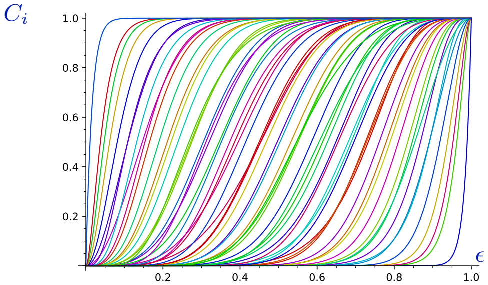
Capacities $C_i = C(W_i)$ of bit channels as a function of ϵ, $N = 32$

Channel ordering on the BEC



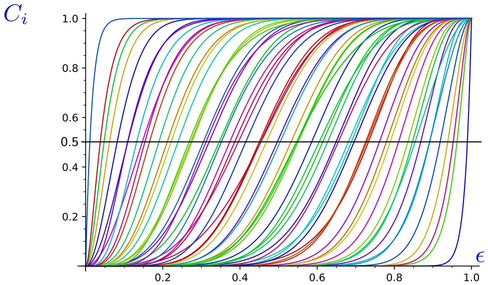
Capacities $C_i = C(W_i)$ of bit channels as a function of ϵ, $N = 64$

Channel ordering on the BEC



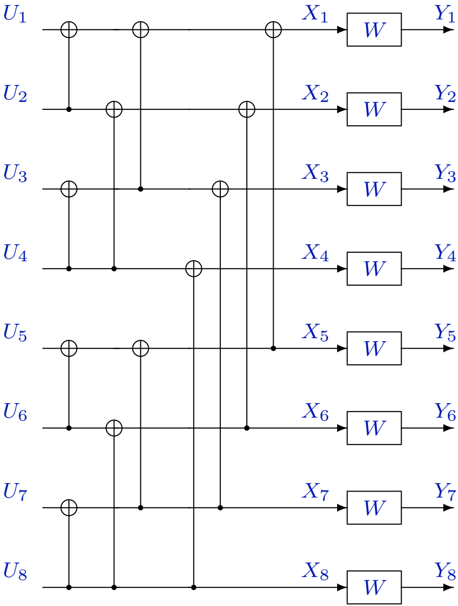
- Order generally depends on ϵ, although parts of it appear fixed and independent of ϵ.

Channel ordering on the BEC



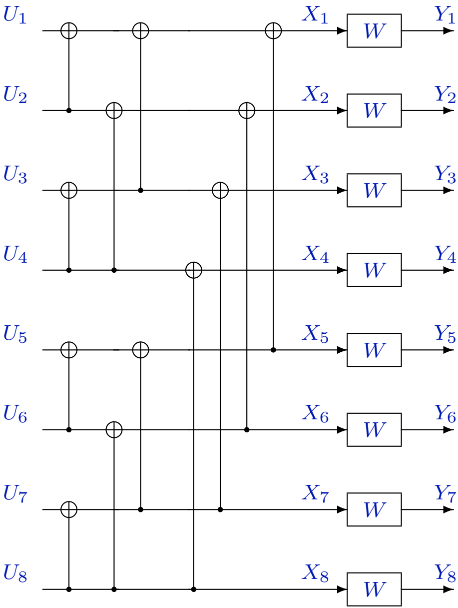
- Order generally depends on ϵ, although parts of it appear fixed and independent of ϵ.
- [Ordentlich and Roth (2019)] show that ordering according to $\alpha_i = C^{-1}(\frac{1}{2})$ (independently of ϵ) still achieves capacity under SCD, although with diminished convergence rates.
- [Wu and Siegel (2019)] further study "universal" partial orders for the BEC and more general channels.

Encoding example: $N = 8$, $K = 4$ on BEC(0.5)



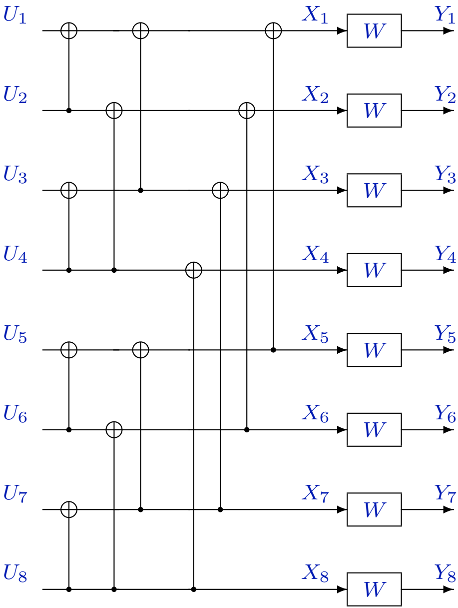
Encoding example: $N = 8,\ K = 4$ on BEC(0.5)

$C(W_i)$	rank
0.0039	8
0.1211	7
0.1914	6
0.6836	4
0.3164	5
0.8086	3
0.8789	2
0.9961	1

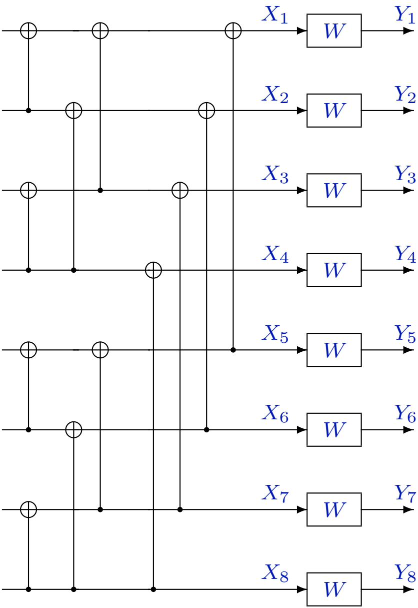


Encoding example: $N = 8,\ K = 4$ on BEC(0.5)

$C(W_i)$	rank	val
0.0039	8	
0.1211	7	
0.1914	6	
0.6836	4	data
0.3164	5	
0.8086	3	data
0.8789	2	data
0.9961	1	data

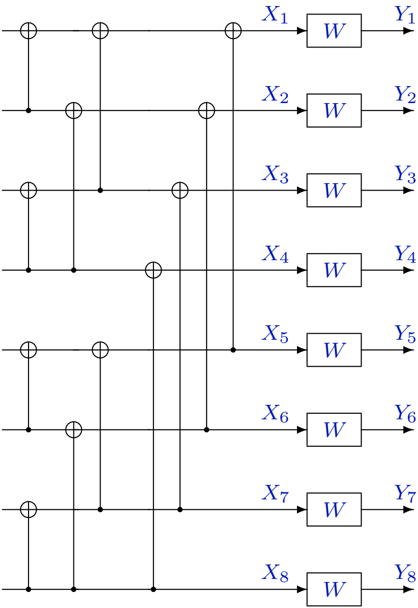


$C(W_i)$	rank	val	
0.0039	8	frozen	U_1 0
0.1211	7	frozen	U_2 0
0.1914	6	frozen	U_3 0
0.6836	4	data	U_4
0.3164	5	frozen	U_5 0
0.8086	3	data	U_6
0.8789	2	data	U_7
0.9961	1	data	U_8



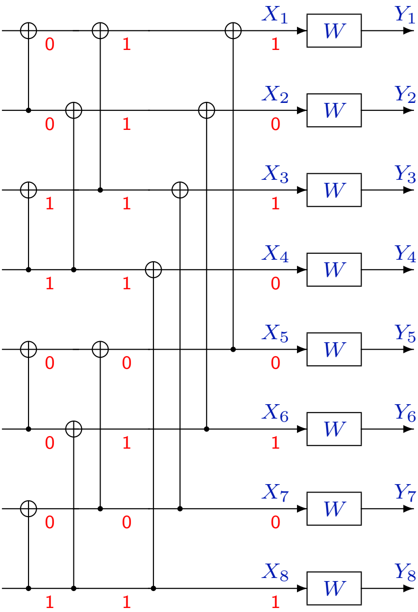
Encoding example: $N = 8,\ K = 4$ on BEC(0.5)

$C(W_i)$	rank	val	U_i
0.0039	8	frozen	U_1 0
0.1211	7	frozen	U_2 0
0.1914	6	frozen	U_3 0
0.6836	4	data	U_4 1
0.3164	5	frozen	U_5 0
0.8086	3	data	U_6 0
0.8789	2	data	U_7 1
0.9961	1	data	U_8 1



Encoding example: $N = 8,\ K = 4$ on BEC(0.5)

$C(W_i)$	rank	val	
0.0039	8	frozen	U_1 0
0.1211	7	frozen	U_2 0
0.1914	6	frozen	U_3 0
0.6836	4	data	U_4 1
0.3164	5	frozen	U_5 0
0.8086	3	data	U_6 0
0.8789	2	data	U_7 1
0.9961	1	data	U_8 1



Successive cancellation decoding (SCD)

This is the original scheme proposed by Arikan.

The receiver generates an estimate $\hat{u}^n$ of the input u^n, as follows.

For $i = 1, 2, \ldots, N$:

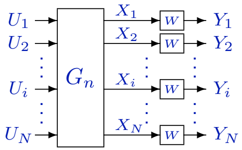
$$\hat{u}_i = \begin{cases} u_i & \text{if } i \in \mathcal{A}^c, \\ g_i(y^N, \hat{u}^{i-1}) & \text{if } i \in \mathcal{A}, \end{cases}$$

where $g_i,\ i \in \mathcal{A}$, are *decision functions*

$$g_i(y^N, \hat{u}^{i-1}) \triangleq \begin{cases} 0 & \text{if } \dfrac{W_i(y^N, \hat{u}^{i-1} | u_i = 0)}{W_i(y^N, \hat{u}^{i-1} | u_i = 1)} \geq 1, \\ 1, & \text{otherwise.} \end{cases}$$

We say that a *decoder block error* occurred if $\hat{u}^N \neq u^N$, or, equivalently $\hat{u}_{\mathcal{A}} \neq u_{\mathcal{A}}$.

Recall:



$W_i : U_i \to (Y^N, U^{i-1})$

Performance of SCD

Error performance.

Theorem

For any rate $R = K/N < C(W)$ *and block-length* N*, the probability of block error for polar codes under successive cancellation decoding is bounded as*

$$P_e(N, R) = o(2^{-\sqrt{N}+o(\sqrt{N})}) \,.$$

Complexity. Here, too the structure of G_N allows for an efficient implementation.

Theorem

The complexity of successive cancellation decoding for polar codes is $O(N \log N)$*.*

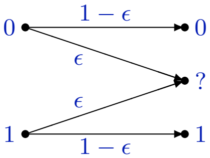
SCD issues

- Compared to ML decoding, SCD is sub-optimal, because it does not take advantage of the knowledge of frozen bits with indices $j > i$ when estimating $\hat{u}_i$. However, the penalty does not prevent SCD from approaching channel capacity.
- For channels other than the BEC, the original SCD computation may still be costly (hidden costs in the complexity of computing precise decisions). Many improvements have been developed, successfully addressing these issues.

A reflection on polar codes and Shannon's paradise

Taking the BEC as an example:

- The channel is *perfect* a fraction $1-\epsilon$ of the times, and *useless* a fraction ϵ of the times.
- Of course, we do not know which times are going to be perfect, and which useless.

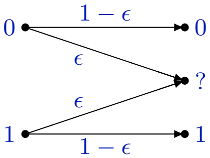


Capacity $C(W) = 1-\epsilon$

A reflection on polar codes and Shannon's paradise

Taking the BEC as an example:

- The channel is *perfect* a fraction $1-\epsilon$ of the times, and *useless* a fraction ϵ of the times.
- Of course, we do not know which times are going to be perfect, and which useless.
- The BEC capacity is $C(W) = 1-\epsilon$, and Shannon tells us we can communicate at a rate arbitrarily close to $1-\epsilon$.

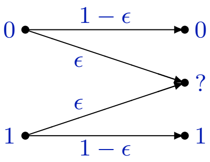


Capacity $C(W) = 1-\epsilon$

A reflection on polar codes and Shannon's paradise

Taking the BEC as an example:

- The channel is *perfect* a fraction $1-\epsilon$ of the times, and *useless* a fraction ϵ of the times.
- Of course, we do not know which times are going to be perfect, and which useless.
- The BEC capacity is $C(W)=1-\epsilon$, and Shannon tells us we can communicate at a rate arbitrarily close to $1-\epsilon$.
- So, Shannon is telling us we can communicate *as if we had a genie telling us which times are going to be perfect, and which useless*, and we could send data only at the perfect times.

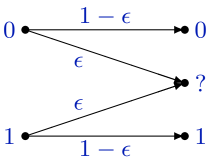


Capacity $C(W)=1-\epsilon$

A reflection on polar codes and Shannon's paradise

Taking the BEC as an example:

- The channel is *perfect* a fraction $1-\epsilon$ of the times, and *useless* a fraction ϵ of the times.
- Of course, we do not know which times are going to be perfect, and which useless.
- The BEC capacity is $C(W) = 1-\epsilon$, and Shannon tells us we can communicate at a rate arbitrarily close to $1-\epsilon$.
- So, Shannon is telling us we can communicate *as if we had a genie telling us which times are going to be perfect, and which useless*, and we could send data only at the perfect times.



Capacity $C(W) = 1-\epsilon$

Polar codes make Shannon's genie real by designing and identifying the perfect channels and the useless ones, and sending data only over the perfect ones.

Polar codes: development

Polar codes have been extensively studied and improved since the original publication. Extensions/improvements include:

- non-binary inputs
- non-symmetric channels (where the *symmetric capacity* can be attained, generally inferior to the full capacity)
- systematic encoding
- concatenated schemes (with CRC and other codes)
- efficient list decoding
- multi-user settings
- applications to source coding
- many improvements in complexity of code construction and encoding/decoding algorithms, enabling the practical application of the codes
- polar codes adopted as part of the 5G standard