Polar Coding
Status and Prospects

Erdal Arıkan

Electrical-Electronics Engineering Department
Bilkent University
Ankara, Turkey

1 August 2011
The IEEE International Symposium on Information Theory
ISIT’2011
Saint Petersburg, Russia
The channel

Let $W : X \rightarrow Y$ be a binary-input discrete memoryless channel.

- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$,
- transition probabilities:
  
  $W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y}$
The channel

Let $W : X \rightarrow Y$ be a binary-input discrete memoryless channel

\[ X \xrightarrow{W} Y \]

- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$,
- transition probabilities:

\[ W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y} \]
The channel

Let $W : X \rightarrow Y$ be a binary-input discrete memoryless channel

\[ X \xrightarrow{W} Y \]

- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$,
- transition probabilities:

\[ W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y} \]
The channel

Let $W : X \rightarrow Y$ be a binary-input discrete memoryless channel

- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$,
- transition probabilities:

$$W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y}$$
Symmetry assumption

Assume that the channel has “input-output symmetry.”
Symmetry assumption

Assume that the channel has “input-output symmetry.”

Examples:
Symmetry assumption

Assume that the channel has “input-output symmetry.”

Examples:

- BSC(\(\epsilon\))
  - Input: 0, 1
  - Output: 0, 1
  - Probabilities:
    - \(1 - \epsilon\) with 0 to 0, 1 to 1
    - \(\epsilon\) with 0 to 1, 1 to 0

- BEC(\(\epsilon\))
  - Input: 0, 1
  - Output: 0, 1
  - Probabilities:
    - \(1 - \epsilon\) with 0 to 0, 1 to 1
    - \(\epsilon\) with 0 to 1, 1 to 0

\(\epsilon\) is a parameter representing the error rate.
Capacity

For channels with input-output symmetry, the capacity is given by

\[ C(W) \overset{\Delta}{=} I(X; Y), \quad \text{with } X \sim \text{unif. } \{0, 1\} \]
Capacity

For channels with input-output symmetry, the capacity is given by

\[ C(W) \overset{\Delta}{=} I(X; Y), \quad \text{with } X \sim \text{unif. } \{0, 1\} \]

Use base-2 logarithms:

\[ 0 \leq C(W) \leq 1 \]
The main idea

- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary $W$ into such extreme channels
The main idea

- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$

- Transform ordinary $W$ into such extreme channels
The main idea

- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary $W$ into such extreme channels
The main idea

- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary $W$ into such extreme channels
The method: aggregate and redistribute capacity

Original channels
(uniform)

$W$

$W$

$W$
The method: aggregate and redistribute capacity

Original channels (uniform)

\[ W \]

\[ \ldots \]

\[ W \]

Vector channel

\[ W_{\text{vec}} \]

Combine
The method: aggregate and redistribute capacity

Original channels (uniform)

\[ W \]

\[ \vdots \]

\[ W \]

\[ W \]

Vector channel

\[ W_{\text{vec}} \]

New channels (polarized)

\[ W_1 \]

\[ \vdots \]

\[ W_{N-1} \]

\[ W_N \]

Combine ➔ Split
Combining

- Begin with $N$ copies of $W$,
- use a 1-1 mapping
  \[ G_N : \{0, 1\}^N \rightarrow \{0, 1\}^N \]
- to create a vector channel
  \[ W_{vec} : U^N \rightarrow Y^N \]
Combining

- Begin with $N$ copies of $W$,
- use a 1-1 mapping

$$G_N : \{0, 1\}^N \rightarrow \{0, 1\}^N$$

- to create a vector channel

$$W_{vec} : U^N \rightarrow Y^N$$
Combining

- Begin with $N$ copies of $W$,
- use a 1-1 mapping
  \[ G_N : \{0, 1\}^N \rightarrow \{0, 1\}^N \]
- to create a vector channel
  \[ W_{vec} : U^N \rightarrow Y^N \]
Conservation of capacity

Combining operation is lossless:

- Take $U_1, \ldots, U_N$ i.i.d. unif. $\{0, 1\}$
- then, $X_1, \ldots, X_N$ i.i.d. unif. $\{0, 1\}$
- and

$$C(W_{vec}) = I(U^N; Y^N)$$
$$= I(X^N; Y^N)$$
$$= NC(W)$$
Conservation of capacity

Combining operation is lossless:

- Take $U_1, \ldots, U_N$ i.i.d. unif. $\{0, 1\}$
- then, $X_1, \ldots, X_N$ i.i.d. unif. $\{0, 1\}$
- and

$$C(W_{\text{vec}}) = I(U^N; Y^N)$$
$$= I(X^N; Y^N)$$
$$= NC(W)$$
Conservation of capacity

Combining operation is lossless:

- Take $U_1, \ldots, U_N$ i.i.d. unif. $\{0, 1\}$
- then, $X_1, \ldots, X_N$ i.i.d. unif. $\{0, 1\}$
- and

$$C(W_{vec}) = I(U^N; Y^N)$$
$$= I(X^N; Y^N)$$
$$= NC(W)$$
Splitting

\[ C(W_{\text{vec}}) = I(U^N; Y^N) \]
\[ C(W_{\text{vec}}) = I(U^N; Y^N) = \sum_{i=1}^{N} I(U_i; Y^N, U^{i-1}) \]
Splitting

\[
C(W_{\text{vec}}) = I(U^N; Y^N) = \sum_{i=1}^{N} I(U_i; Y^N, U^{i-1})
\]

Define bit-channels

\[
W_i : U_i \rightarrow (Y^N, U^{i-1})
\]
Splitting

\[ C(W_{\text{vec}}) = I(U^N; Y^N) = \sum_{i=1}^{N} I(U_i; Y^N, U^{i-1}) = \sum_{i=1}^{N} C(W_i) \]

Define bit-channels

\[ W_i : U_i \rightarrow (Y^N, U^{i-1}) \]
Polarization is commonplace

- Polarization is the rule not the exception
  - A random permutation
    \[ G_N : \{0, 1\}^N \rightarrow \{0, 1\}^N \]
    is a good polarizer with high probability
  - Equivalent to Shannon’s random coding approach
Polarization is commonplace

- Polarization is the rule not the exception
- A random permutation
  
  \[ G_N : \{0, 1\}^N \rightarrow \{0, 1\}^N \]

  is a good polarizer with high probability
- Equivalent to Shannon’s random coding approach
Polarization is commonplace

- Polarization is the rule not the exception
- A random permutation
  \[ G_N : \{0, 1\}^N \rightarrow \{0, 1\}^N \]
  is a good polarizer with high probability
- Equivalent to Shannon’s random coding approach
Random polarizers: stepwise, isotropic
Random polarizers: stepwise, isotropic

Isotropy: any redistribution order is as good as any other.
The complexity issue

- Random polarizers lack structure, too complex to implement
- Need a low-complexity polarizer
- May sacrifice stepwise, isotropic properties of random polarizers in return for less complexity
The complexity issue

- Random polarizers lack structure, too complex to implement
- Need a low-complexity polarizer
- May sacrifice stepwise, isotropic properties of random polarizers in return for less complexity
The complexity issue

- Random polarizers lack structure, too complex to implement
- Need a low-complexity polarizer
- May sacrifice stepwise, isotropic properties of random polarizers in return for less complexity
Basic module for a low-complexity scheme

Combine two copies of $W$

\[
\begin{align*}
W & \quad Y_1 \\
X_1 & \\
W & \quad Y_2 \\
X_2 &
\end{align*}
\]
Basic module for a low-complexity scheme

Combine two copies of $W$
Basic module for a low-complexity scheme

Combine two copies of $W$

\[ U_1 \rightarrow (Y_1, Y_2) \quad \text{and} \quad U_2 \rightarrow (Y_1, Y_2, U_1) \]
The first bit-channel $\mathcal{W}_1$ 

\[ \mathcal{W}_1 : U_1 \rightarrow (Y_1, Y_2) \]
The first bit-channel $W_1$

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

$C(W_1) = I(U_1; Y_1, Y_2)$
The second bit-channel $W_2$

$W_2 : U_2 \rightarrow (Y_1, Y_2, U_1)$
The second bit-channel $W_2$

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

$C(W_2) = I(U_2; Y_1, Y_2, U_1)$
Capacity conserved but redistributed unevenly

Conservation:

\[ C(W_1) + C(W_2) = 2C(W) \]

Extremization:

\[ C(W_1) \leq C(W) \leq C(W_2) \]

with equality iff \( C(W) \) equals 0 or 1.
Capacity conserved but redistributed unevenly

Conservation:

\[ C(W_1) + C(W_2) = 2C(W) \]

Extremization:

\[ C(W_1) \leq C(W) \leq C(W_2) \]

with equality iff \( C(W) \) equals 0 or 1.
The two channels created by the basic transform

\[(W, W) \rightarrow (W_1, W_2)\]

will be denoted also as

\[W^- = W_1 \quad \text{and} \quad W^+ = W_2\]
Notation

The two channels created by the basic transform

$$(W, W) \rightarrow (W_1, W_2)$$

will be denoted also as

$$W^- = W_1 \quad \text{and} \quad W^+ = W_2$$

Likewise, we write $W^{--}, W^{-+}$ for descendants of $W^-$; and $W^{+-}, W^{++}$ for descendants of $W^+$. 
For the size-4 construction
... duplicate the basic transform
... obtain a pair of $W^-$ and $W^+$ each
... apply basic transform on each pair
... decode in the indicated order

\[ U_1 \rightarrow \neg \neg W^- \rightarrow W^- \rightarrow W^+ \rightarrow W^+ \]

\[ U_3 \rightarrow \neg \neg W^+ \rightarrow W^+ \rightarrow W^- \rightarrow W^- \]

\[ U_2 \rightarrow \neg \neg W^- \rightarrow W^- \rightarrow W^+ \rightarrow W^+ \]

\[ U_4 \rightarrow \neg \neg W^+ \rightarrow W^+ \rightarrow W^- \rightarrow W^- \]
... obtain the four new bit-channels

$U_1$ --- $W^{--}$

$U_3$ --- $W^{++}$

$U_2$ --- $W^{--}$

$U_4$ --- $W^{++}$
Overall size-4 construction
“Rewire” for standard-form size-4 construction

\[ U_1 \quad + \quad X_1 \quad W \quad Y_1 \]
\[ U_2 \quad + \quad X_2 \quad W \quad Y_2 \]
\[ U_3 \quad + \quad X_3 \quad W \quad Y_3 \]
\[ U_4 \quad + \quad X_4 \quad W \quad Y_4 \]
Size 8 construction
Demonstration of polarization

Polarization is easy to analyze when $W$ is a BEC.

If $W$ is a BEC($\epsilon$), then so are $W^-$ and $W^+$, with erasure probabilities

$\epsilon^- \overset{\Delta}{=} 2\epsilon - \epsilon^2$

and

$\epsilon^+ \overset{\Delta}{=} \epsilon^2$

respectively.
Demonstration of polarization

Polarization is easy to analyze when $W$ is a BEC.

If $W$ is a BEC($\epsilon$), then so are $W^-$ and $W^+$, with erasure probabilities

$$\epsilon^- \triangleq 2\epsilon - \epsilon^2$$

and

$$\epsilon^+ \triangleq \epsilon^2$$

respectively.
Demonstration of polarization

Polarization is easy to analyze when $W$ is a BEC.

If $W$ is a BEC($\epsilon$), then so are $W^-$ and $W^+$, with erasure probabilities

$$\epsilon^- \equiv 2\epsilon - \epsilon^2$$

and

$$\epsilon^+ \equiv \epsilon^2$$

respectively.
Polarization for BEC($\frac{1}{2}$): $N = 16$
Polarization for BEC($\frac{1}{2}$): $N = 32$
Polarization for BEC($\frac{1}{2}$): $N = 64$

![Graph showing the capacity of bit channels for BEC with $N = 64$. The x-axis represents the bit channel index, and the y-axis represents the capacity. The graph contains data points illustrating the capacity distribution across different bit channels.]
Polarization for BEC($\frac{1}{2}$): $N = 128$
Polarization for BEC($\frac{1}{2}$): $N = 256$
Polarization for BEC($\frac{1}{2}$): $N = 512
Polarization for BEC($\frac{1}{2}$): $N = 1024$
Polarization martingale

\[ C(W) \]
Polarization martingale

\[ C(W_1) \quad C(W) \quad C(W_2) \]
Polarization martingale

\[ C(W) \xrightarrow{} C(W_1) \xrightarrow{} C(W_2) \xrightarrow{} \cdots \]

\[ C(W_{++}) \xrightarrow{} C(W_{+-}) \xrightarrow{} C(W_{-+}) \xrightarrow{} C(W_{--}) \]
Polarization martingale

\[ C(W) \]

\[ C(W_1) \]

\[ C(W_{++}) \]

\[ C(W_{+-}) \]

\[ C(W_{-+}) \]

\[ C(W_{--}) \]

\[ C(W_2) \]
Polarization martingale

\[
C(W) \rightarrow C(W^-) \rightarrow C(W^+) \rightarrow C(W_2) \rightarrow C(W^{++}) \rightarrow 1
\]
Polarization martingale
Polarization martingale
Polarization martingale
Polarization martingale
Polarization martingale
Theorem (Polarization, A. 2007)

The bit-channel capacities \( \{ C(W_i) \} \) polarize: for any \( \delta \in (0, 1) \), as the construction size \( N \) grows

\[
\left[ \frac{\text{no. channels with } C(W_i) > 1 - \delta}{N} \right] \rightarrow C(W)
\]

and

\[
\left[ \frac{\text{no. channels with } C(W_i) < \delta}{N} \right] \rightarrow 1 - C(W)
\]

Theorem (Rate of polarization, A. and Telatar (2008))

Above theorem holds with \( \delta \approx 2^{-\sqrt{N}} \).
Theorem (Polarization, A. 2007)

The bit-channel capacities $\{C(W_i)\}$ polarize: for any $\delta \in (0, 1)$, as the construction size $N$ grows

$$\left[ \frac{\text{no. channels with } C(W_i) > 1 - \delta}{N} \right] \longrightarrow C(W)$$

and

$$\left[ \frac{\text{no. channels with } C(W_i) < \delta}{N} \right] \longrightarrow 1 - C(W)$$

Theorem (Rate of polarization, A. and Telatar (2008))

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

Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate 1/2

<table>
<thead>
<tr>
<th>$C(W_i)$</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0039</td>
<td>8</td>
</tr>
<tr>
<td>0.1211</td>
<td>7</td>
</tr>
<tr>
<td>0.1914</td>
<td>6</td>
</tr>
<tr>
<td>0.6836</td>
<td>4</td>
</tr>
<tr>
<td>0.3164</td>
<td>5</td>
</tr>
<tr>
<td>0.8086</td>
<td>3</td>
</tr>
<tr>
<td>0.8789</td>
<td>2</td>
</tr>
<tr>
<td>0.9961</td>
<td>1</td>
</tr>
</tbody>
</table>
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

<table>
<thead>
<tr>
<th>$C(W_i)$</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0039</td>
<td>8</td>
</tr>
<tr>
<td>0.1211</td>
<td>7</td>
</tr>
<tr>
<td>0.1914</td>
<td>6</td>
</tr>
<tr>
<td>0.6836</td>
<td>4</td>
</tr>
<tr>
<td>0.3164</td>
<td>5</td>
</tr>
<tr>
<td>0.8086</td>
<td>3</td>
</tr>
<tr>
<td>0.8789</td>
<td>2</td>
</tr>
</tbody>
</table>
| 0.9961   | 1    | **data**
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate 1/2

<table>
<thead>
<tr>
<th>$C(W_i)$</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0039</td>
<td>8</td>
</tr>
<tr>
<td>0.1211</td>
<td>7</td>
</tr>
<tr>
<td>0.1914</td>
<td>6</td>
</tr>
<tr>
<td>0.6836</td>
<td>4</td>
</tr>
<tr>
<td>0.3164</td>
<td>5</td>
</tr>
<tr>
<td>0.8086</td>
<td>3</td>
</tr>
<tr>
<td>0.8789</td>
<td>2</td>
</tr>
<tr>
<td>0.9961</td>
<td>1</td>
</tr>
</tbody>
</table>
Polar code example: \( W = \text{BEC}(\frac{1}{2}), \ N = 8, \ \text{rate} \ 1/2 \)

<table>
<thead>
<tr>
<th>( C(W_i) )</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0039</td>
<td>8</td>
</tr>
<tr>
<td>0.1211</td>
<td>7</td>
</tr>
<tr>
<td>0.1914</td>
<td>6</td>
</tr>
<tr>
<td>0.6836</td>
<td>4</td>
</tr>
<tr>
<td>0.3164</td>
<td>5</td>
</tr>
<tr>
<td>0.8086</td>
<td>3</td>
</tr>
<tr>
<td>0.8789</td>
<td>2</td>
</tr>
<tr>
<td>0.9961</td>
<td>1</td>
</tr>
</tbody>
</table>
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

<table>
<thead>
<tr>
<th>$C(W_i)$</th>
<th>Rank</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0039</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>0.1211</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>0.1914</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>0.6836</td>
<td>4</td>
<td>data</td>
</tr>
<tr>
<td>0.3164</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>0.8086</td>
<td>3</td>
<td>data</td>
</tr>
<tr>
<td>0.8789</td>
<td>2</td>
<td>data</td>
</tr>
<tr>
<td>0.9961</td>
<td>1</td>
<td>data</td>
</tr>
</tbody>
</table>
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate 1/2

<table>
<thead>
<tr>
<th>$C(W_i)$</th>
<th>Rank</th>
<th>State</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0039</td>
<td>8</td>
<td>frozen</td>
<td>$U_1$</td>
</tr>
<tr>
<td>0.1211</td>
<td>7</td>
<td>frozen</td>
<td>$U_2$</td>
</tr>
<tr>
<td>0.1914</td>
<td>6</td>
<td>frozen</td>
<td>$U_3$</td>
</tr>
<tr>
<td>0.6836</td>
<td>4</td>
<td>data</td>
<td>$U_4$</td>
</tr>
<tr>
<td>0.3164</td>
<td>5</td>
<td>frozen</td>
<td>$U_5$</td>
</tr>
<tr>
<td>0.8086</td>
<td>3</td>
<td>data</td>
<td>$U_6$</td>
</tr>
<tr>
<td>0.8789</td>
<td>2</td>
<td>data</td>
<td>$U_7$</td>
</tr>
<tr>
<td>0.9961</td>
<td>1</td>
<td>data</td>
<td>$U_8$</td>
</tr>
</tbody>
</table>
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

<table>
<thead>
<tr>
<th>$C(W_i)$</th>
<th>Rank</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0039</td>
<td>8</td>
<td>frozen</td>
<td>0</td>
</tr>
<tr>
<td>0.1211</td>
<td>7</td>
<td>frozen</td>
<td>0</td>
</tr>
<tr>
<td>0.1914</td>
<td>6</td>
<td>frozen</td>
<td>0</td>
</tr>
<tr>
<td>0.6836</td>
<td>4</td>
<td>data</td>
<td>$U_4$</td>
</tr>
<tr>
<td>0.3164</td>
<td>5</td>
<td>frozen</td>
<td>0</td>
</tr>
<tr>
<td>0.8086</td>
<td>3</td>
<td>data</td>
<td>$U_6$</td>
</tr>
<tr>
<td>0.8789</td>
<td>2</td>
<td>data</td>
<td>$U_7$</td>
</tr>
<tr>
<td>0.9961</td>
<td>1</td>
<td>data</td>
<td>$U_8$</td>
</tr>
</tbody>
</table>
Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT’2011)
Construction complexity

- An $O(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $O(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT’2011)
Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT’2011)
Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation

- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking

- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT’2011)
Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT’2011)
Encoding complexity

Encoding complexity for polar coding is $O(N \log N)$. 
Encoding: an example

```
<table>
<thead>
<tr>
<th>frozen</th>
<th>0</th>
<th>W</th>
<th>Y_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>frozen</td>
<td>0</td>
<td>W</td>
<td>Y_2</td>
</tr>
<tr>
<td>frozen</td>
<td>0</td>
<td>W</td>
<td>Y_3</td>
</tr>
<tr>
<td>free</td>
<td>1</td>
<td>W</td>
<td>Y_4</td>
</tr>
<tr>
<td>frozen</td>
<td>0</td>
<td>W</td>
<td>Y_5</td>
</tr>
<tr>
<td>free</td>
<td>1</td>
<td>W</td>
<td>Y_6</td>
</tr>
<tr>
<td>free</td>
<td>0</td>
<td>W</td>
<td>Y_7</td>
</tr>
<tr>
<td>free</td>
<td>1</td>
<td>W</td>
<td>Y_8</td>
</tr>
</tbody>
</table>
```
Encoding: an example

```
frozen  0  0  \[W\] \rightarrow Y_1
frozen  0  0  \[W\] \rightarrow Y_2
frozen  0  1  \[W\] \rightarrow Y_3
  free  1  1  \[W\] \rightarrow Y_4
frozen  0  1  \[W\] \rightarrow Y_5
  free  1  1  \[W\] \rightarrow Y_6
  free  0  1  \[W\] \rightarrow Y_7
  free  1  1  \[W\] \rightarrow Y_8
```
Encoding: an example
Encoding: an example
### Successive cancellation decoding complexity

**(A. 2007)**

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

(A. 2007)
Complexity of successive cancellation decoding for polar codes is $O(N \log N)$.

Earlier work on similar decoders:

- Kabatiansky (1990)
- Schnabl and Bossert (1996)
- Dumer and co-authors (from 1990s)
- Burnashev and Dumer (2006-2009)
Performance of SC decoder

(A. and Telatar, 2008)

For any rate $R < C(W)$ and block-length $N$, the probability of frame error for polar codes under SC decoding is bounded roughly as

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

- Prior result (A. 2007): $P_e(N, R) = o \left(N^{-1/4}\right)$.
- Latest result: A rate-dependent refinement of the “square-root” bound has been given by Tanaka & Mori (2010) and Hassani & Urbanke (2010).
Performance of SC decoder

(A. and Telatar, 2008)

For any rate $R < C(W)$ and block-length $N$, the probability of frame error for polar codes under SC decoding is bounded roughly as

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

- Prior result (A. 2007): $P_e(N, R) = o\left(N^{-1/4}\right)$.
- Latest result: A rate-dependent refinement of the “square-root” bound has been given by Tanaka & Mori (2010) and Hassani & Urbanke (2010)
Performance of SC decoder

(A. and Telatar, 2008)

For any rate $R < C(W)$ and block-length $N$, the probability of frame error for polar codes under SC decoding is bounded roughly as

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

- Prior result (A. 2007): $P_e(N, R) = o \left( N^{-1/4} \right)$.
- Latest result: A rate-dependent refinement of the “square-root” bound has been given by Tanaka & Mori (2010) and Hassani & Urbanke (2010).
Polar coding summary

Given $W$, $N = 2^n$, and $R < C(W)$, a polar code with these parameters has

- construction complexity $\mathcal{O}(N)$ (conjecture),
- encoding complexity $\approx N \log N$,
- decoding complexity $\approx N \log N$,
- and frame error probability $P_e(N, R) \approx o(2^{-\sqrt{N}})$
Polar coding summary

Given $W, N = 2^n$, and $R < C(W)$, a polar code with these parameters has

- construction complexity $O(N)$ (conjecture),
- encoding complexity $\approx N \log N$,
- decoding complexity $\approx N \log N$,
- and frame error probability $P_e(N, R) \approx o(2^{-\sqrt{N}})$
Polar coding summary

Given $W$, $N = 2^n$, and $R < C(W)$, a polar code with these parameters has

- construction complexity $\mathcal{O}(N)$ (conjecture),
- encoding complexity $\approx N \log N$,
- decoding complexity $\approx N \log N$,
- and frame error probability $P_e(N, R) \approx o(2^{-\sqrt{N}})$.
Polar coding summary

Given $W$, $N = 2^n$, and $R < C(W)$, a polar code with these parameters has

- construction complexity $\mathcal{O}(N)$ (conjecture),
- encoding complexity $\approx N \log N$,
- decoding complexity $\approx N \log N$,
- and frame error probability $P_e(N, R) \approx o(2^{-\sqrt{N}})$.
Polar coding in other contexts

- Source coding (lossless)
- Source coding in the presence of memory
- Lossy source coding
- Slepian-Wolf problem
- Wyner-Ziv problem
- Gelfand-Pinsker problem
- MAC
- Degraded-broadcast channel
- Wyner wiretap channel
- Randomness extraction
- ...

...
## Channel-coding scenarios

<table>
<thead>
<tr>
<th>Topic</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-user $q$-ary channels</td>
<td>Şaşoğlu, Telatar, A. (2009)</td>
</tr>
<tr>
<td>Multi-access channels</td>
<td>Şaşoğlu, Telatar, Yeh (2010)</td>
</tr>
<tr>
<td>$m$-user MAC</td>
<td>Abbe and Telatar (2010)</td>
</tr>
<tr>
<td>Wyner wiretap channel</td>
<td>Mahdavifar and Vardy (2009)</td>
</tr>
<tr>
<td>&quot;</td>
<td>Hof and Shamai (2010)</td>
</tr>
<tr>
<td>&quot;</td>
<td>Koyluoglu and El Gamal (2010)</td>
</tr>
<tr>
<td>&quot;</td>
<td>Andersson et al. (2010)</td>
</tr>
<tr>
<td>Relay channel</td>
<td>Andersson et al. (2010)</td>
</tr>
<tr>
<td>&quot;</td>
<td>Blasco-Serrano et al. (2010)</td>
</tr>
<tr>
<td>&quot;</td>
<td>Karzand (2011)</td>
</tr>
<tr>
<td>Compound channel coding</td>
<td>Hassani, Korada, Urbanke (2009)</td>
</tr>
</tbody>
</table>
# Source-coding scenarios

<table>
<thead>
<tr>
<th>Topic</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lossless source coding</td>
<td>Hussami, Korada, Urbanke (2009)</td>
</tr>
<tr>
<td>Rate-distortion coding</td>
<td>Korada and Urbanke (2009)</td>
</tr>
<tr>
<td>$q$-ary lossless source coding</td>
<td>Karzand and Telatar (2010)</td>
</tr>
<tr>
<td>Direct source polarization</td>
<td>A. (2010)</td>
</tr>
<tr>
<td>Universal polar coding</td>
<td>Abbe (2010)</td>
</tr>
<tr>
<td>Sparse recovery</td>
<td>Abbe (2010)</td>
</tr>
<tr>
<td>Randomness extraction</td>
<td>Abbe (2011)</td>
</tr>
<tr>
<td>Ergodic source polarization</td>
<td>Şaşoğlu (2011)</td>
</tr>
</tbody>
</table>
Scenarios with side-information

<table>
<thead>
<tr>
<th>Topic</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gelfand-Pinsker coding</td>
<td>Korada and Urbanke (2009)</td>
</tr>
<tr>
<td>Slepian-Wolf coding</td>
<td>Hussami, Korada, Urbanke (2009)</td>
</tr>
</tbody>
</table>
**Generalized polarization schemes**

$q$: alphabet size  
$\ell$: dimension of basic transform (kernel)  
$E$: rate of polarization exponent

<table>
<thead>
<tr>
<th>$q$</th>
<th>$\ell$</th>
<th>Exponent $E$</th>
<th>Kernel</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2 to 15</td>
<td>$\leq 1/2$</td>
<td>Any linear</td>
<td>KSU (2009)</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>0.51828</td>
<td>BCH</td>
<td>KSU (2009)</td>
</tr>
<tr>
<td>2</td>
<td>31</td>
<td>0.52643</td>
<td>BCH</td>
<td>KSU (2009)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.573120</td>
<td>Reed-Solomon</td>
<td>MT (2010)</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>0.50193</td>
<td>Nonlinear</td>
<td>PSL (2011)</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>0.50773</td>
<td>Nonlinear</td>
<td>PSL (2011)</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>0.52742</td>
<td>Nonlinear</td>
<td>PSL (2011)</td>
</tr>
</tbody>
</table>

KSU: Korada, Şasoğlu, Urbanke  
MT: Mori and Tanaka  
PSL: Presman, Shapira, Litsyn
Performance comparison: Polar vs. Turbo

Turbo code
- WiMAX CTC
- Duobinary, memory 3
- QAM over AWGN channel
- Gray mapping
- BICM
- Simulator: “Coded Modulation Library”

Polar code
- Standard construction
- Successive cancellation decoding
- QAM over AWGN channel
- Natural mapping
- Multi-level PAM
- PAM over AWGN channel
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits \((b_1, b_2, b_3)\)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order \(b_3, b_2, b_1\)
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits \((b_1, b_2, b_3)\)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order \(b_3, b_2, b_1\)
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits \((b_1, b_2, b_3)\)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order \(b_3, b_2, b_1\)
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits \((b_1, b_2, b_3)\)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order \(b_3, b_2, b_1\)
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits \((b_1, b_2, b_3)\)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order \(b_3, b_2, b_1\)
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits \((b_1, b_2, b_3)\)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order \(b_3, b_2, b_1\)
Multi-layering jump-starts polarization
4-QAM, Rate 1/2

![Graph showing FER vs EbNo for various coding schemes: Polar(512,256), Polar(1024,512), CTC(480,240), CTC(960,480) for 4-QAM.](image-url)
16-QAM, Rate 3/4

EbNo (dB) vs FER for different codes:
- Polar(512,384) 16-QAM
- CTC(192,144) 16-QAM
- CTC(384,288) 16-QAM
- CTC(576,432) 16-QAM
64-QAM, Rate 5/6

The graph shows the performance comparison of different encoding schemes for 64-QAM modulation with a rate of 5/6. The x-axis represents the EbNo (Energy per bit over noise power spectral density) in dB, while the y-axis represents the FER (Frame Error Rate) on a logarithmic scale. The curves indicate the performance of Polar coding schemes with different dimensions and the CTC (Convolutional Trellis Coding) scheme.
Complexity comparison: 64-QAM, Rate 5/6

Average decoding time in milliseconds per codeword (ms/cw)

<table>
<thead>
<tr>
<th>$E_b/N_0$</th>
<th>CTC(576,432)</th>
<th>Polar(768,640)</th>
<th>Polar(384,320)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 dB</td>
<td>6.23</td>
<td>0.92</td>
<td>0.48</td>
</tr>
<tr>
<td>11 dB</td>
<td>1.83</td>
<td>1.01</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Both decoders implemented as MATLAB mex functions. Polar decoder is a successive cancellation decoder. CTC decoder is a public domain decoder (CML). Profiling done by MATLAB Profiler. Iteration limit for CTC decoder was 10; average no of iterations was 10 at 10 dB and 3.3 at 11 dB. CTC decoder used a linear approximation to log-MAP while polar decoder used exact log-MAP.
Complexity comparison: 64-QAM, Rate 5/6

Average decoding time in milliseconds per codeword (ms/cw)

<table>
<thead>
<tr>
<th>$E_b/N_0$</th>
<th>CTC(576,432)</th>
<th>Polar(768,640)</th>
<th>Polar(384,320)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 dB</td>
<td>6.23</td>
<td>0.92</td>
<td>0.48</td>
</tr>
<tr>
<td>11 dB</td>
<td>1.83</td>
<td>1.01</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Polar codes show a complexity advantage against CTC codes.

Both decoders implemented as MATLAB mex functions. Polar decoder is a successive cancellation decoder. CTC decoder is a public domain decoder (CML). Profiling done by MATLAB Profiler. Iteration limit for CTC decoder was 10; average no of iterations was 10 at 10 dB and 3.3 at 11 dB. CTC decoder used a linear approximation to log-MAP while polar decoder used exact log-MAP.
Performance improvement for polar codes

- Concatenation to improve minimum distance
- List decoding to improve SC decoder performance
Performance improvement for polar codes

- Concatenation to improve minimum distance
- List decoding to improve SC decoder performance
Concatenation

<table>
<thead>
<tr>
<th>Method</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block turbo coding with polar constituents</td>
<td>AKMOP (2009)</td>
</tr>
<tr>
<td>Generalized concatenated coding with polar inner</td>
<td>AM (2009)</td>
</tr>
<tr>
<td>Reed-Solomon outer, polar inner</td>
<td>BJE (2010)</td>
</tr>
<tr>
<td>Polar outer, block inner</td>
<td>SH (2010)</td>
</tr>
<tr>
<td>Polar outer, LDPC inner</td>
<td>EP (ISIT’2011)</td>
</tr>
</tbody>
</table>

AKMOP: A., Kim, Markarian, Özgür, Poyraz
GCC: A., Markarian
BJE: Bakshi, Jaggi, and Effros
SH: Seidl and Huber
EP: Eslami and Pishro-Nik
Tal-Vardy list decoder for polar codes

- First produce $L$ candidate decisions
- Pick the most likely word from the list
- Complexity $O(LN \log N)$
Tal-Vardy list decoder for polar codes

- First produce $L$ candidate decisions
- Pick the most likely word from the list
- Complexity $O(LN \log N)$
Tal-Vardy list decoder for polar codes

- First produce $L$ candidate decisions
- Pick the most likely word from the list
- Complexity $O(LN \log N)$
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$. 

![Graph showing bit error rate vs. signal-to-noise ratio for different list sizes.](image)
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$. 
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$. 

![Graph showing bit error rate vs. signal-to-noise ratio for different list sizes $L$.](image)
**Tal-Vardy list decoder performance**

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$. 

![Graph showing bit error rate versus signal-to-noise ratio for different list sizes ($L = 1, 2, 4, 8, 16$).]
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$. 
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$. 

![Graph showing the performance of Tal-Vardy list decoder. The x-axis represents the signal-to-noise ratio in dB, and the y-axis represents the bit error rate on a logarithmic scale. The graph shows different curves for various list sizes $L$, with the ML bound represented by a dashed line.](image-url)
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$.

List-of-$L$ performance quickly approaches ML performance!
Tal-Vardy list decoder with CRC

- Same decoder as before but data contains a built-in CRC
- Selection made by CRC and relative likelihood
Tal-Vardy list decoder with CRC

- Same decoder as before but data contains a built-in CRC
- Selection made by CRC and relative likelihood
Tal-Vardy list decoder with CRC

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$. 

![Graph showing performance comparison of different decoding methods.]

- **Successive cancellation**
- **List-decoding ($L = 32$)**
- **Polar ML bound**
Tal-Vardy list decoder with CRC

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$. 

![Graph showing bit error rate vs. signal-to-noise ratio with various decoding techniques compared to different channel models and bounds.](image_url)
Tal-Vardy list decoder with CRC

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size $L$.

Polar codes (+CRC) achieve state-of-the-art performance!
Hardware implementation of polar codes

▶ Advantages
  ▶ Regular structure simplifies resource reuse
  ▶ Lack of randomness helps avoid memory conflicts

▶ Disadvantages
  ▶ High latency: $\mathcal{O}(N)$
  ▶ Throughput bottleneck: 1/2 bits per clock-period

Hardware implementation of polar codes

- Advantages
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts

- Disadvantages
  - High latency: $O(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

Hardware implementation of polar codes

- Advantages
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts

- Disadvantages
  - High latency: $O(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

Hardware implementation of polar codes

- **Advantages**
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts

- **Disadvantages**
  - High latency: $\mathcal{O}(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

Hardware implementation of polar codes

- **Advantages**
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts

- **Disadvantages**
  - High latency: $O(N)$
  - Throughput bottleneck: $1/2$ bits per clock-period

Hardware implementation of polar codes

▶ Advantages
  ▶ Regular structure simplifies resource reuse
  ▶ Lack of randomness helps avoid memory conflicts

▶ Disadvantages
  ▶ High latency: \( O(N) \)
  ▶ Throughput bottleneck: 1/2 bits per clock-period

Summary

- Polarization is a commonplace phenomenon – almost unavoidable
- Polar codes are low-complexity methods designed to exploit polarization for achieving Shannon limits
- Polar codes with some help from other methods perform competitively with the state-of-the-art codes in terms of complexity and performance
Summary

- Polarization is a commonplace phenomenon – almost unavoidable
- Polar codes are low-complexity methods designed to exploit polarization for achieving Shannon limits
- Polar codes with some help from other methods perform competitively with the state-of-the-art codes in terms of complexity and performance
Summary

- Polarization is a commonplace phenomenon – almost unavoidable
- Polar codes are low-complexity methods designed to exploit polarization for achieving Shannon limits
- Polar codes with some help from other methods perform competitively with the state-of-the-art codes in terms of complexity and performance
Acknowledgements

Research sponsors

Special thanks to Emre Telatar, Alex Vardy, Ido Tal, and Warren Gross for sharing ideas, software and slides, and to Matthew Valenti for Coded Modulation Library.
Thank you!