Полярные коды

Кирилл Андреев

k.andreev@skoltech.ru

Сколковский институт науки и технологий (Сколтех)

Современные методы теории информации, оптимизации и управления

Сочи, Россия

2–23 августа, 2020

Содержание

Полярные коды в двоичном канале

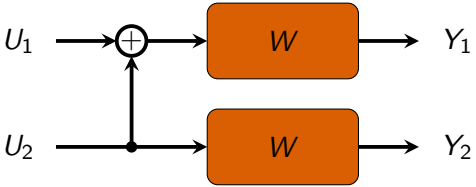
- Существование помехоустойчивых кодов, достигающих пропускной способности канала, доказано Шенноном в 1948 году
- Однако, алгоритм построения таких кодов не был описан
- Полярные коды – первая явная кодовая конструкция, достигающая пропускной способности симметричного канала

E. Arikan
Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels // IEEE Trans. on Inf. Theory, vol. 55, no. 7, 2009.

Содержание

Полярное преобразование

- Рассмотрим две копии одного и того же двоичного канала $W : \mathbb{F}_2 \to \mathcal{Y}$
- Зададим $X_1 = U_1 \oplus U_2, \quad X_2 = U_2$
- $(U_1, U_2) \sim U(\mathbb{F}_2) \Rightarrow X_1, X_2 \sim U(\mathbb{F}_2)$



1. Полярное преобразование сохраняет взаимную информацию

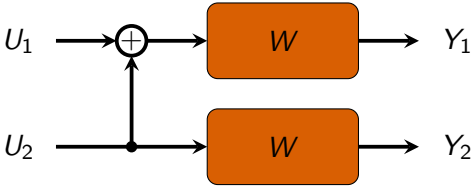
$$
\begin{aligned}
2I(W) = I(X_1X_2; Y_1Y_2) = I(U_1U_2; Y_1Y_2) &= I(U_1; X_1X_2) + I(U_2, X_1X_2|U_1) \\
&= I(U_1; X_1X_2) + I(U_2, X_1X_2U_1) \\
&= I(W^-) + I(W^+)
\end{aligned}
$$

2. Происходит «поляризация» канала

$$I(W^-) \le I(W) \le I(W^+)$$

Декодирование при полярном преобразовании

- $W^- : U_1 \to Y_1 Y_2$: декодирование U_1 по известным декодеру выходам Y_1 и Y_2
- $W^+ : U_2 \to Y_1 Y_2 U_1$: значение U_1 недоступно декодеру



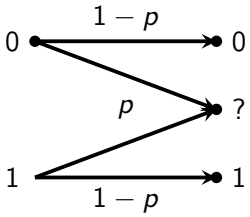
Рассмотрим два варианта декодера:

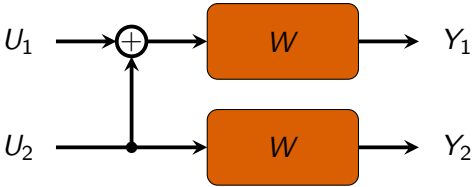
$$\begin{aligned} \tilde{U}_1 &= \varphi_1(Y_1, Y_2) \\ \tilde{U}_2 &= \varphi_2(Y_1, Y_2, U_1) \end{aligned} \qquad \begin{aligned} \hat{U}_1 &= \varphi_1(Y_1, Y_2) \\ \hat{U}_2 &= \varphi_2(Y_1, Y_2, \hat{U}_1) \end{aligned}$$

- Возможно ли при дальнейшем анализе декодера заменить U_1 на $\hat{U}_1$?
- Рассмотрим ошибку декодирования блока $U^2 = (U_1, U_2)$

$$\tilde{U}^2 \neq U^2 \iff \hat{U}^2 \neq U^2$$

Двоичный канал со стиранием – BEC

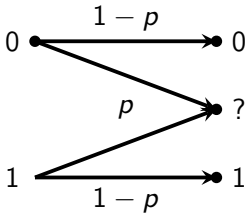


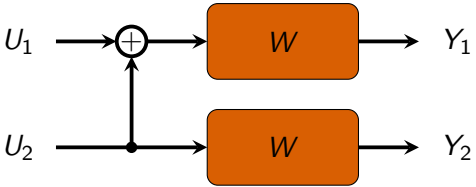


$W^{-}: U_1 \to Y_1 Y_2$ – это **BEC** $\left(\mathbf{2p - p^2}\right)$

$$(Y_1, Y_2) = \begin{cases} (\ U_1 \oplus U_2\ ,\ U_2\ ), & (1-p)^2, \\ (\ ?\ ,\ U_2\ ), & p(1-p), \\ (\ U_1 \oplus U_2\ ,\ ?\ ), & (1-p)p, \\ (\ ?\ ,\ ?\ ), & p^2, \end{cases}$$

Двоичный канал со стиранием – BEC





$$W^+ : U_2 \to Y_1 Y_2 U_1 \text{ – это } \mathbf{BEC}\left(\mathbf{p^2}\right)$$

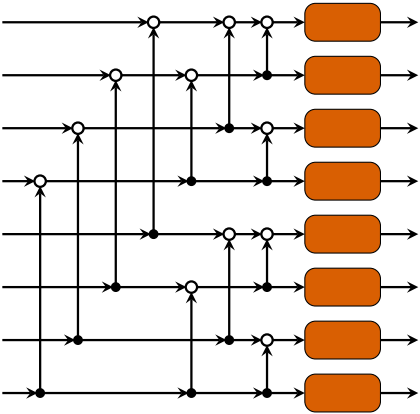
$$(Y_1, Y_2, U_1) = \begin{cases} (\quad U_1 \oplus U_2 \quad , \quad U_2 \quad , U_1), & (1-p)^2, \\ (\quad ? \quad , \quad U_2 \quad , U_1), & p(1-p), \\ (\quad U_1 \oplus U_2 \quad , \quad ? \quad , U_1), & (1-p)p, \\ (\quad ? \quad , \quad ? \quad , U_1), & p^2, \end{cases}$$

Двоичный канал со стиранием – BEC

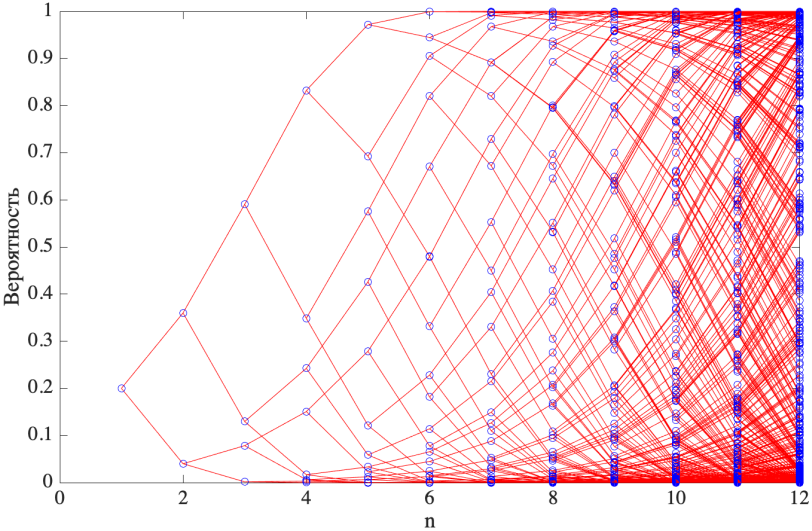
- Результат полярного преобразования – «расщепление» двух идентичных каналов $BEC(p)$ на $BEC(p^+)$ и $BEC(p^-)$

$$p^+ = p^2, \quad p^- = 2p - p^2, p^+ \leq p \leq p^-$$

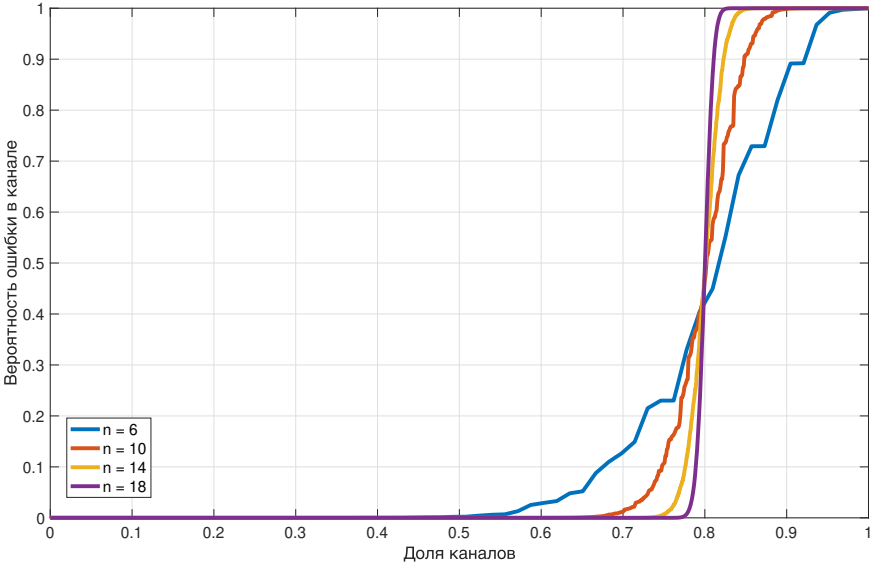
- Можно ли повторить процедуру поляризации канала?



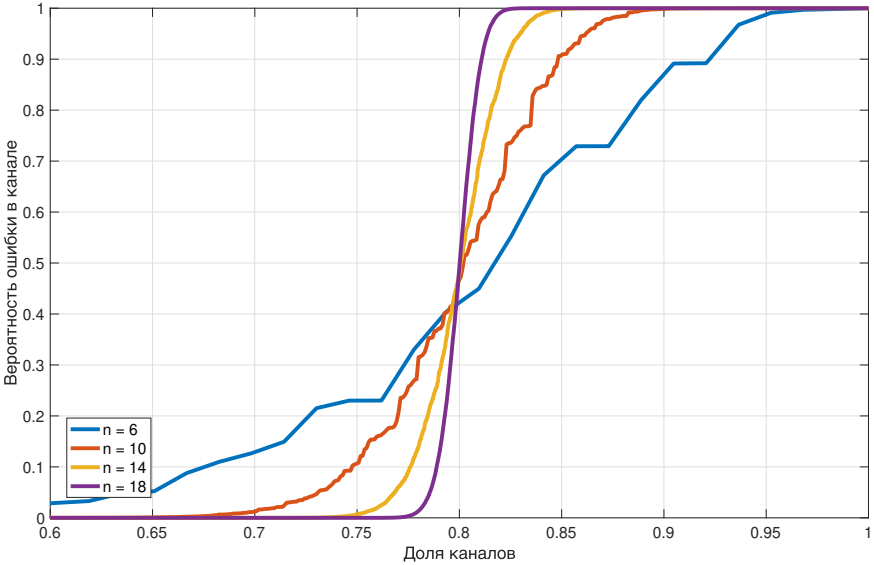
Поляризация двоичного канала со стиранием ($p = 0.2$), $N = 2^n$



Поляризация двоичного канала со стиранием ($p = 0.2$), $N = 2^n$



Поляризация двоичного канала со стиранием ($p = 0.2$), $N = 2^n$



Поляризация двоичного канала со стиранием ($p = 0.2$), $N = 2^n$

- Для части каналов вероятность ошибок близка к нулю, для части каналов – к единице
- Доля каналов с $p \in (0, 1)$ уменьшается с ростом n
- Пропускная способность канала со стираниями: $C = 1 - p$
- Преобразование всех виртуальных каналов или в «идеальные» ($p = 0$), или в «бесполезные» ($p = 1$)

Поляризация двоичного канала со стиранием ($p = 0.2$), $N = 2^n$

- Для части каналов вероятность ошибок близка к нулю, для части каналов – к единице
- Доля каналов с $p \in (0, 1)$ уменьшается с ростом n
- Пропускная способность канала со стираниями: $C = 1 - p$
- Преобразование всех виртуальных каналов или в «идеальные» ($p = 0$), или в «бесполезные» ($p = 1$)

Теорема (О поляризации канала. Arikan, 2009)

При $n \to \infty$ происходит поляризация канала: вероятность ошибки стремится или к нулю, или к единице, причем доля каналов с нулевой ошибкой стремится к пропускной способности канала

Skoltech
Skolkovo Institute of Science and Technology

Содержание

Порождающая матрица полярного кода

Произведение и степень Кронекера

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots \\ a_{2,1}B & a_{2,2}B & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Порождающая матрица полярного кода

Произведение и степень Кронекера

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots \\ a_{2,1}B & a_{2,2}B & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

$$A^{\otimes n} = A \otimes A^{\otimes(n-1)} = A^{\otimes(n-1)} \otimes A$$

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Порождающая матрица полярного кода

Произведение и степень Кронекера

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots \\ a_{2,1}B & a_{2,2}B & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

$$A^{\otimes n} = A \otimes A^{\otimes(n-1)} = A^{\otimes(n-1)} \otimes A$$

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad G_2^{\otimes 2} = \begin{bmatrix} G_2 & 0 \\ G_2 & G_2 \end{bmatrix}$$

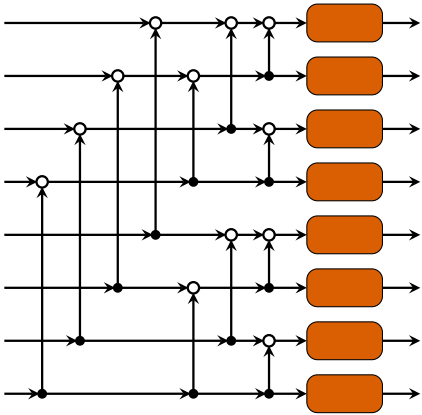
Применение полярных кодов на практике

- Передача информации осуществляется только по «хорошим» подканалам, для плохих каналов значение передаваемых бит фиксируется, или *замораживается*
- Выбор замороженных бит строится для различных каналов
  - Непосредстенным вычислением вероятностей ошибок (BEC, BSC)
  - С помощью моделирования
- Полярный код является линейным кодом $\Rightarrow$ для построения необходимо получить порождающую матрицу кода

Skoltech
Skolkovo Institute of Science and Technology

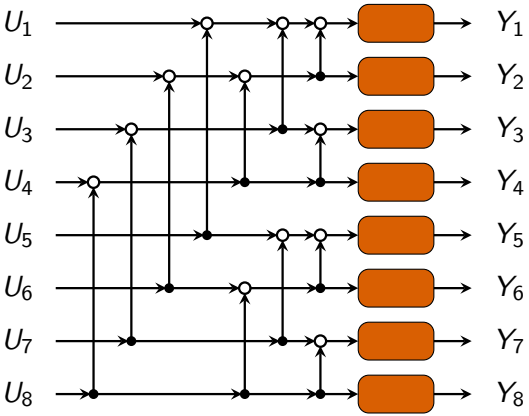
Порождающая матрица полярного кода

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



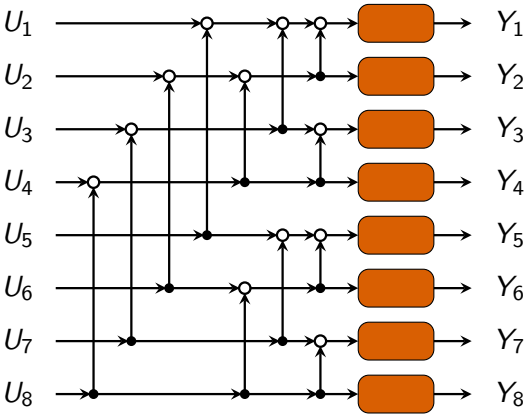
- Строки матрицы G, соответствующие замороженным битам, удаляются

Порождающая матрица полярного кода $W = BEC(0.5)$, $N = 8$, $R = 1/2$



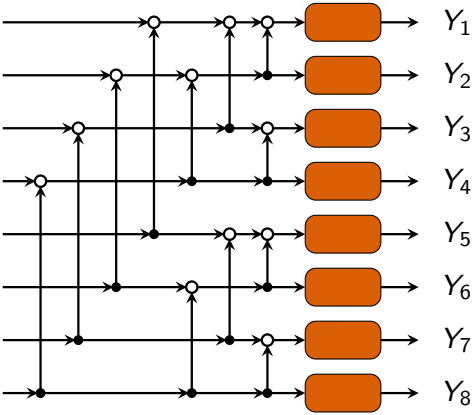
Порождающая матрица полярного кода $W = BEC(0.5)$, $N = 8$, $R = 1/2$

$I(W_i)$
0.0039
0.3164
0.1914
0.8789
0.1211
0.8086
0.6836
0.9961



Порождающая матрица полярного кода $W = BEC(0.5)$, $N = 8$, $R = 1/2$

$I(W_i)$		
0.0039	frozen	U_1
0.3164	frozen	U_2
0.1914	frozen	U_3
0.8789	data	U_4
0.1211	frozen	U_5
0.8086	data	U_6
0.6836	data	U_7
0.9961	data	U_8



Кодироание полярных кодов

- Длина блока полярного кода $N = 2^n$
- Сложность алгоритма кодирования $O\left(N \log N\right)$[1]

[1] Умножение на порождающую матрицу полярного кода дает большую сложность!

Кодироание полярных кодов

- Длина блока полярного кода $N = 2^n$
- Сложность алгоритма кодирования $O\left(N \log N\right)$[1]
  - Полярное преобразование можно представить в виде графа с $N\left(1 + \log N\right)$ вершин
  - Процесс кодирования начинается с первого «слоя» с последовательным применением операции

$$(\mathbf{U}_1, \mathbf{U}_2) \to (\mathbf{U}_1 \oplus \mathbf{U}_2, \mathbf{U}_2)$$

[1] Умножение на порождающую матрицу полярного кода дает большую сложность!

Кодироание полярных кодов

- Длина блока полярного кода $N = 2^n$
- Сложность алгоритма кодирования $O\left(N \log N\right)$[1]
  - Полярное преобразование можно представить в виде графа с $N\left(1 + \log N\right)$ вершин
  - Процесс кодирования начинается с первого «слоя» с последовательным применением операции

$$(\mathbf{U}_1, \mathbf{U}_2) \to (\mathbf{U}_1 \oplus \mathbf{U}_2, \mathbf{U}_2)$$

- Объем необходимой памяти: $O(N)$
- Время выполнения: $O\left(N \log N\right)$

[1] Умножение на порождающую матрицу полярного кода дает большую сложность!

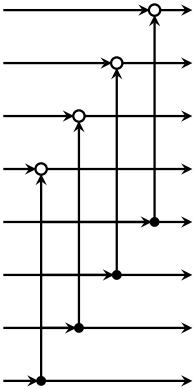
Кодирование полярных кодов

$I(W_i)$		
0.0039	frozen	0
0.3164	frozen	0
0.1914	frozen	0
0.8789	data	1
0.1211	frozen	0
0.8086	data	1
0.6836	data	0
0.9961	data	1

Skoltech
Skolkovo Institute of Science and Technology

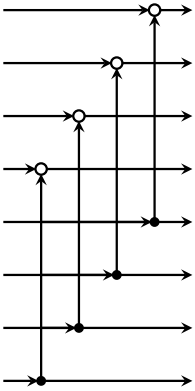
Кодирование полярных кодов

$I(W_i)$		
0.0039	frozen	0
0.3164	frozen	0
0.1914	frozen	0
0.8789	data	1
0.1211	frozen	0
0.8086	data	1
0.6836	data	0
0.9961	data	1



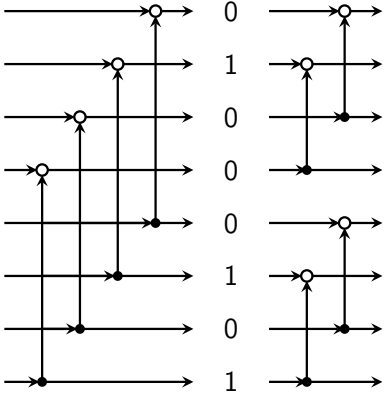
Кодирование полярных кодов

$I(W_i)$			
0.0039	frozen	0	0
0.3164	frozen	0	1
0.1914	frozen	0	0
0.8789	data	1	0
0.1211	frozen	0	0
0.8086	data	1	1
0.6836	data	0	0
0.9961	data	1	1



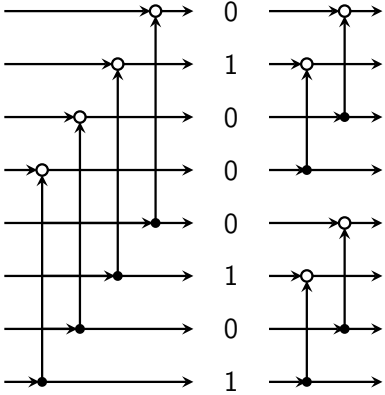
Кодирование полярных кодов

$I(W_i)$				
0.0039	frozen	0	0	
0.3164	frozen	0	1	
0.1914	frozen	0	0	
0.8789	data	1	0	
0.1211	frozen	0	0	
0.8086	data	1	1	
0.6836	data	0	0	
0.9961	data	1	1	



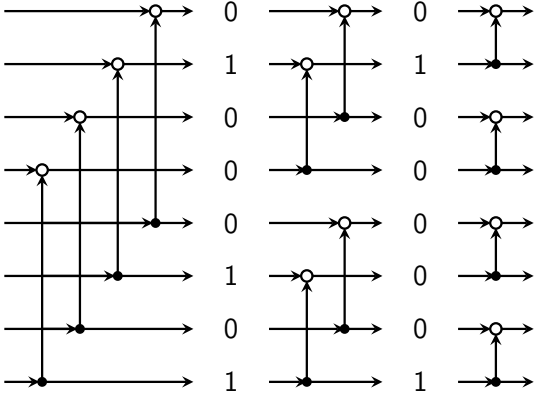
Кодирование полярных кодов

$I(W_i)$					
0.0039	frozen	0	0	0	0
0.3164	frozen	0	1	1	1
0.1914	frozen	0	0	0	0
0.8789	data	1	0	0	0
0.1211	frozen	0	0	0	0
0.8086	data	1	1	0	0
0.6836	data	0	0	0	0
0.9961	data	1	1	1	1



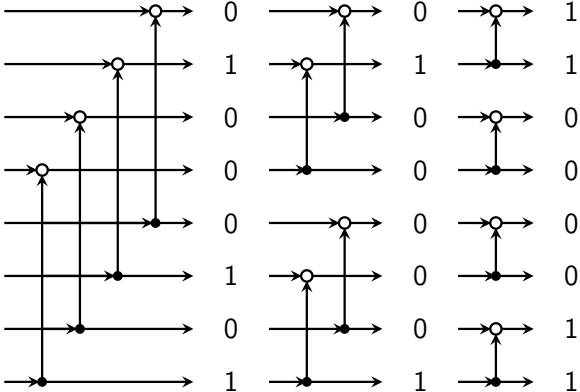
Кодирование полярных кодов

$I(W_i)$					
0.0039	frozen	0	0	0	
0.3164	frozen	0	1	1	
0.1914	frozen	0	0	0	
0.8789	data	1	0	0	
0.1211	frozen	0	0	0	
0.8086	data	1	1	0	
0.6836	data	0	0	0	
0.9961	data	1	1	1	

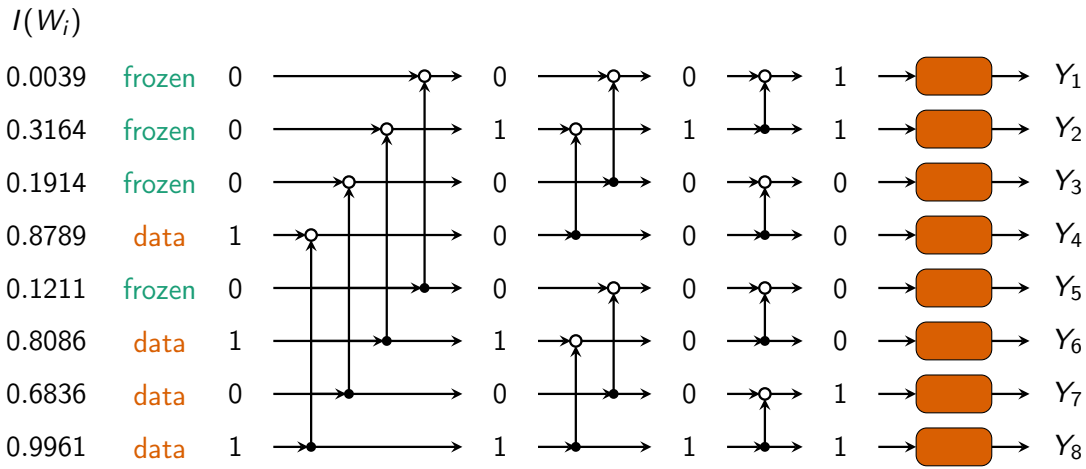


Кодирование полярных кодов

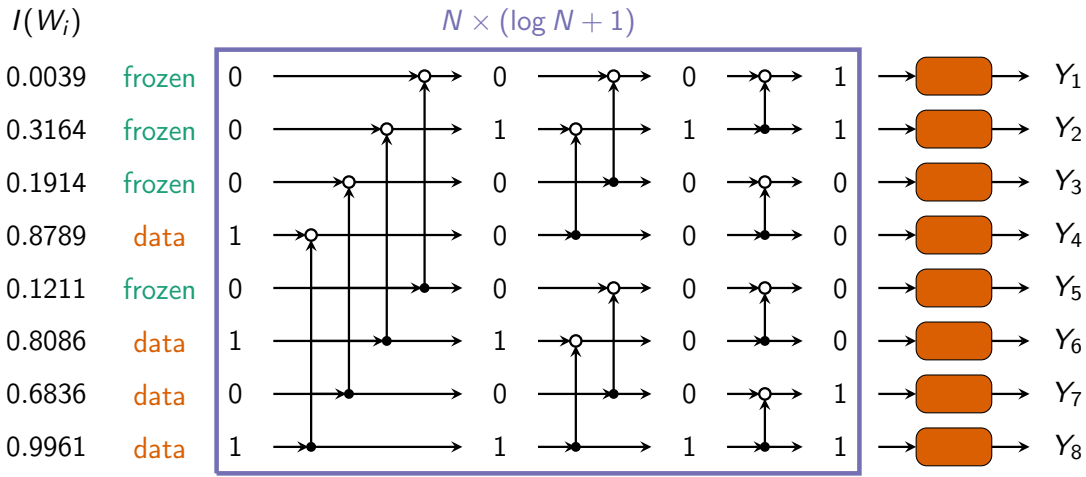
$I(W_i)$							
0.0039	frozen	0	0	0	1
0.3164	frozen	0	1	1	1
0.1914	frozen	0	0	0	0
0.8789	data	1	0	0	0
0.1211	frozen	0	0	0	0
0.8086	data	1	1	0	0
0.6836	data	0	0	0	1
0.9961	data	1	1	1	1



Кодирование полярных кодов



Кодирование полярных кодов



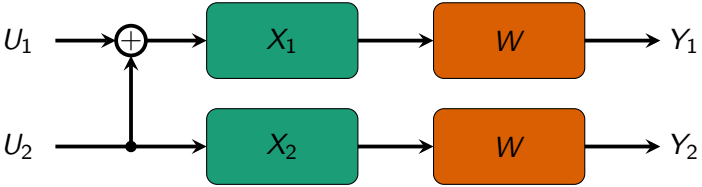
Кодирование полярных кодов

```
function x = polar_transform(u)
if (size(u, 2) == 1)
    x = u;
else
    u1u2 = mod(u(:, 1:2:end) + u(:, 2:2:end), 2);
    u2 = u(:, 2:2:end);
    x = [polar_transform(u1u2), polar_transform(u2)];
end
end
```

Henry D. Pfister
A Brief Introduction to Polar Codes // Supplemental Material for Advanced Channel Coding

Содержание

Декодирование методом последовательных исключений, случай 2×2



- Выход канала: $p_1 = P\left(X_1 = 1 | Y_1\right), \quad p_2 = P\left(X_2 = 1 | Y_2\right)$
- Декодируем символ U_1

$$\begin{aligned} P(U_1 = 1 | U_2, Y_1, Y_2) &= P(U_1 = 1 | U_2 = 0, Y_1, Y_2) + P(U_1 = 1 | U_2 = 0, Y_1, Y_2) \\ &= P(X_1 = 0, X_2 = 1 | Y_1, Y_2) + P(X_2 = 0, X_1 = 1 | Y_1, Y_2) \\ &= p_1(1 - p_2) + p_2(1 - p_1) \end{aligned}$$

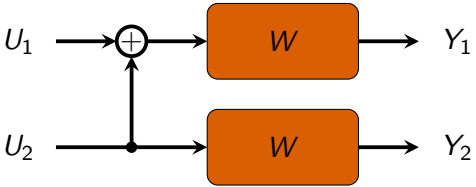
- Декодируем символ U_2 в предположении, что $U_1 = 0$

$$P(U_2 = 1 | U_1 = 0, Y_1, Y_2) = \frac{p_1 p_2}{p_1 p_2 + (1 - p_1)(1 - p_2)}$$

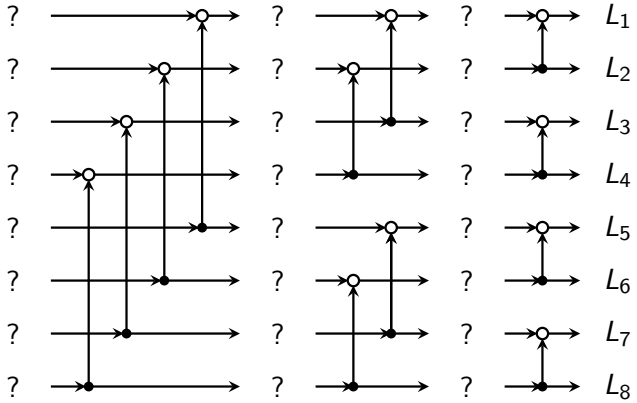
Декодирование методом последовательных исключений, случай 2×2

$$LLR = \log \frac{1-p}{p}$$

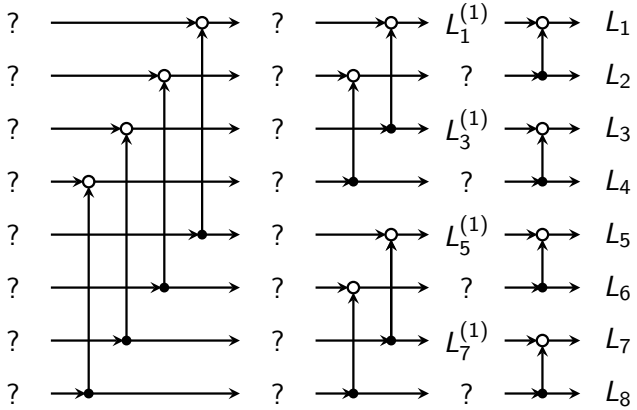
```
function l = cnop_llr(l1, l2)
l = 2 * atanh(tanh(l1 / 2) .* tanh(l2 / 2));
end

function l = vnop_llr(l1, l2, b1)
l = ((-1) .^ b1) .* l1 + l2;
end
```



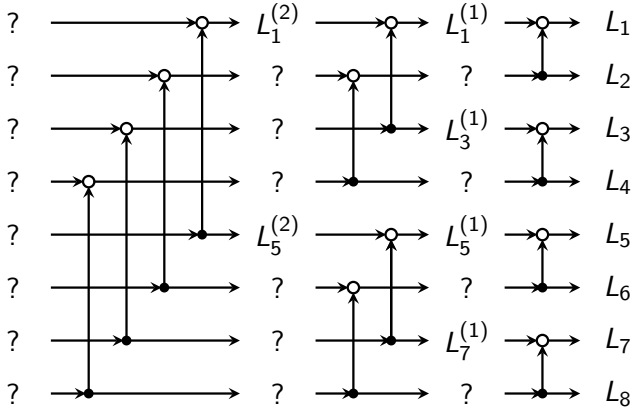
Декодирование полярного кода в общем случае



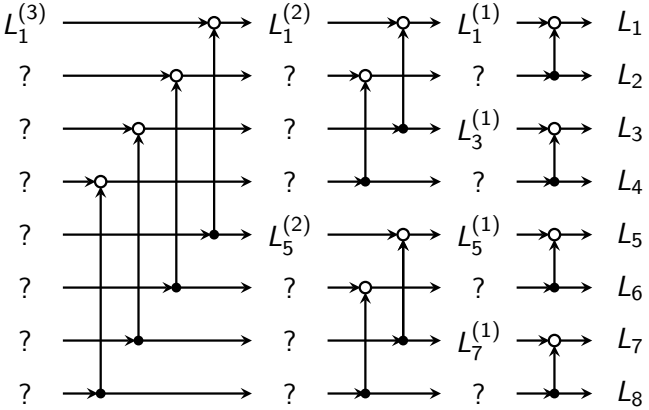
Декодирование полярного кода в общем случае



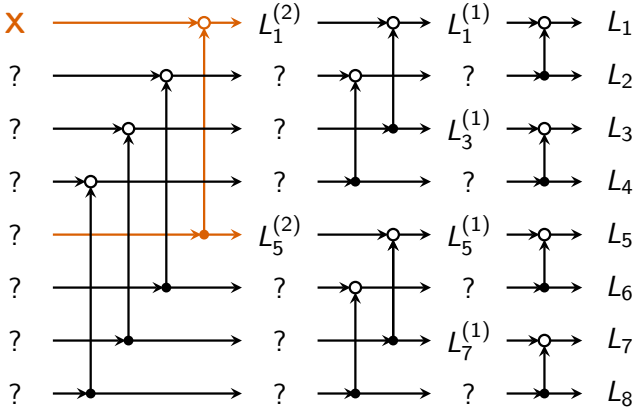
Декодирование полярного кода в общем случае



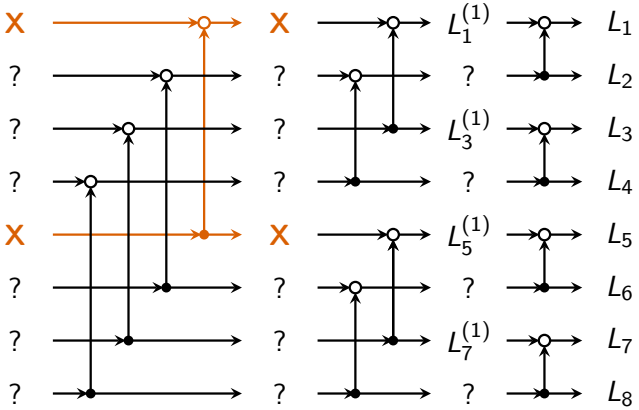
Декодирование полярного кода в общем случае



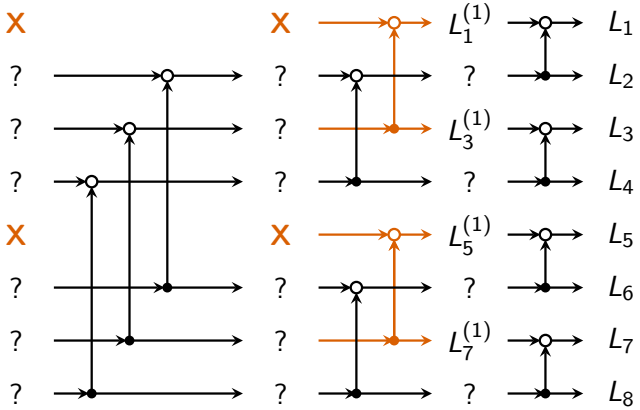
Декодирование полярного кода в общем случае



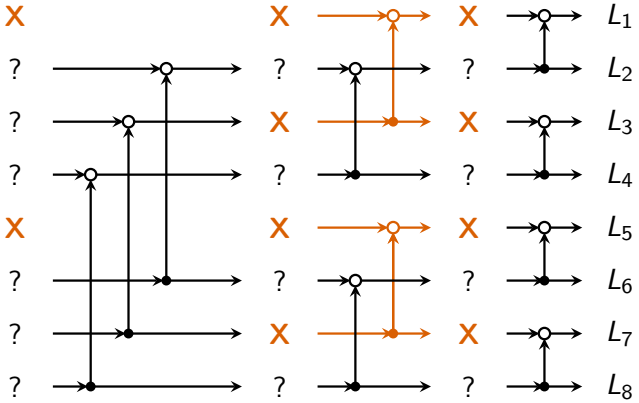
Декодирование полярного кода в общем случае



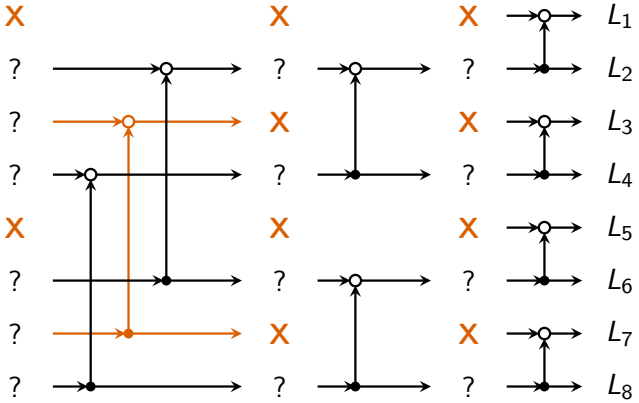
Декодирование полярного кода в общем случае



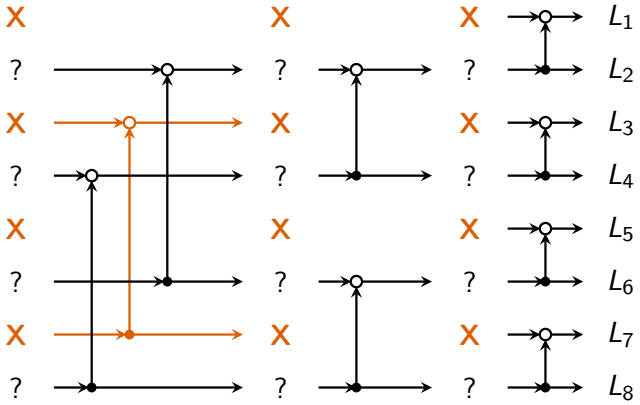
Декодирование полярного кода в общем случае



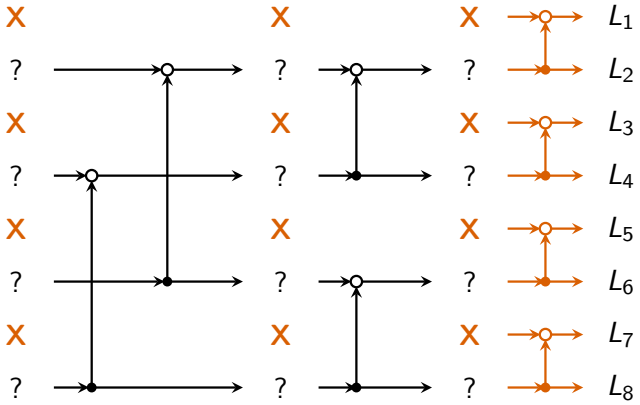
Декодирование полярного кода в общем случае



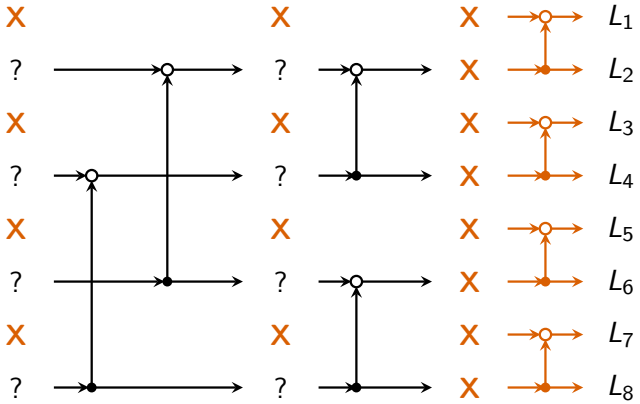
Декодирование полярного кода в общем случае



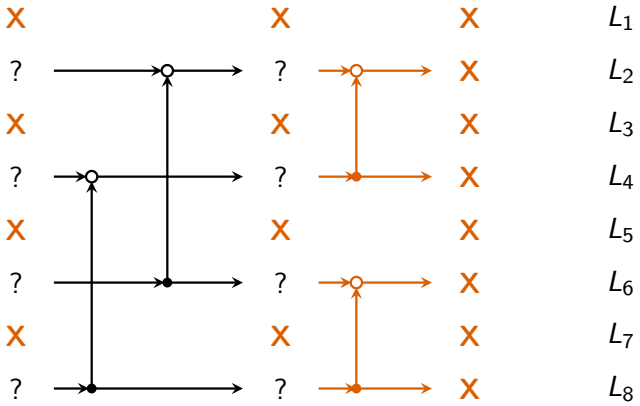
Декодирование полярного кода в общем случае



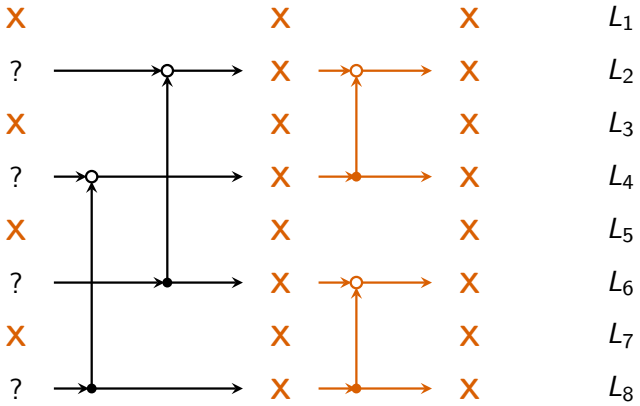
Декодирование полярного кода в общем случае



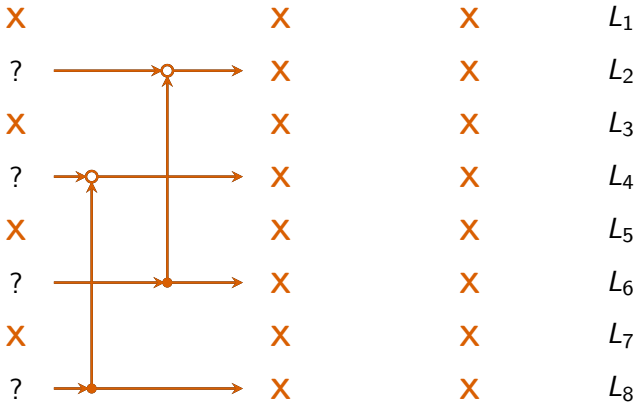
Декодирование полярного кода в общем случае



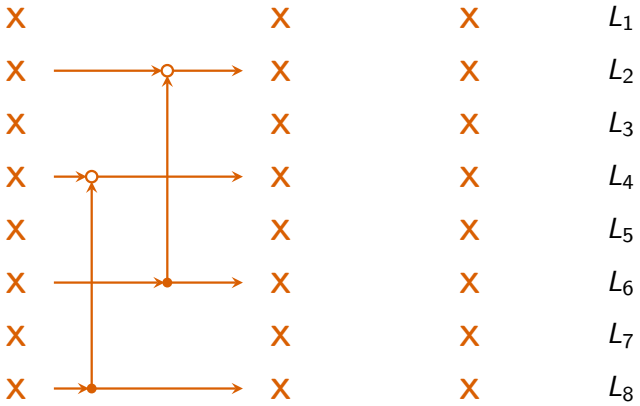
Декодирование полярного кода в общем случае



Декодирование полярного кода в общем случае



Декодирование полярного кода в общем случае



Декодирование полярного кода в общем случае

X	X	X	L_1
X	X	X	L_2
X	X	X	L_3
X	X	X	L_4
X	X	X	L_5
X	X	X	L_6
X	X	X	L_7
X	X	X	L_8

Комментарии

- В каждый момент времени может быть декодирован как минимум один бит – алгоритм называют методом последовательных исключений (Successive cancellation decoding)
- Алгоритм декодирования задает обход графа полярного преобразования и распространения сообщений на этом графе
- Эффективный способ обхода этого графа представлен рекурсивной реализацией декодера

Рекурсивный декодер

```
function cwd_hat = polar_sc_recursive_llr(llr_in, f)
N = size(llr_in, 2);
if (N == 1)
    cwd_hat = (f == 0) * (llr_in < 0);
else
    llr_u = cnop(llr_in(1:2:end), llr_in(2:2:end));
    cwd_u = polar_sc_recursive_llr(llr_u, f(1:(N / 2)));
    llr_l = vnop(llr_in(1:2:end), llr_in(2:2:end), cwd_u);
    cwd_l = polar_sc_recursive_llr(llr_l, f((N / 2 + 1):end));
    cwd_hat = reshape([mod(cwd_u + cwd_l, 2); cwd_l], 1, []);
end
end
```

Метод последовательных исключений

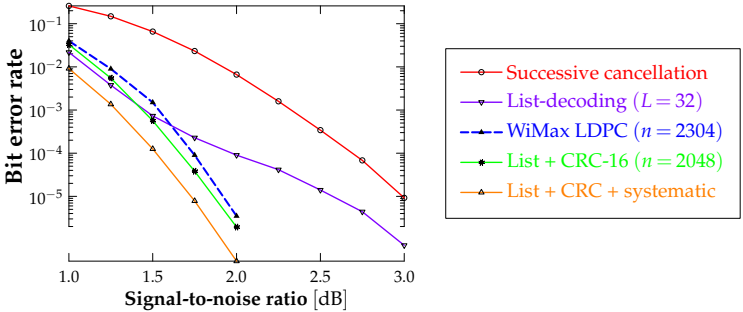
- Доказано, что с помощью полярного преобразования можно построить коды, достигающие пропускной способности симметричного канала
- Метод последовательных исключений – простой алгоритм декодирования
  - Время декодирования $O(N \log N)$
  - Параллельное выполнение невозможно – проблема для длинных кодов

Насколько хорош метод последовательных исключений?

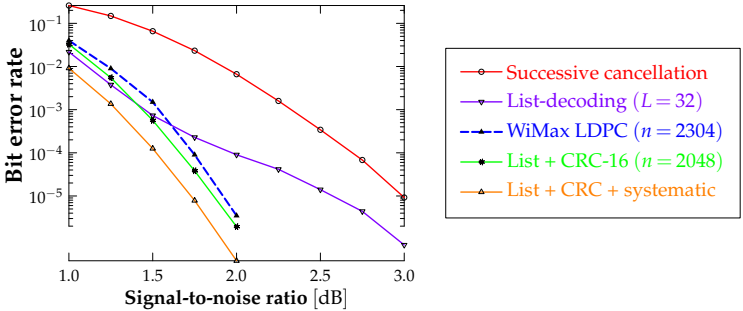
Метод последовательных исключений

- Доказано, что с помощью полярного преобразования можно построить коды, достигающие пропускной способности симметричного канала
- Метод последовательных исключений – простой алгоритм декодирования
  - Время декодирования $O(N \log N)$
  - Параллельное выполнение невозможно – проблема для длинных кодов

Насколько хорош метод последовательных исключений?



Метод последовательных исключений со списком



- Метод последовательных исключений – единственное жесткое решение $\Rightarrow$ для каждого декодируемого бита можно создать две гипотезы
- на шаге k количество гипотез будет 2^k $\Rightarrow$ ограничим размер этого списка, сохраняя только L наиболее вероятных
- Для выбора из списка информационное слово содержит дополнительные проверочные биты (CRC)

Описание проекта

Декодирование полярных кодов с помощью глубоких нейронных сетей

- Проверочная матрица полярного кода имеет высокую плотность проверок
- Использование графа полярного преобразования позволяет получить низкоплотностную матрицу, по которой строится нейросетевой декодер
- Заменить части графа Таннера на полносвязные слои, используя метод последовательных исключений и свойство

$$(\mathbf{U}_1, \mathbf{U}_1 \oplus \mathbf{U}_2)$$

Содержание

Теорема Жегалкина

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- Совершенная дизъюнктивная нормальная форма

$$\bigvee_{\sigma : f(\sigma)=1} x_1^{\sigma_1} \dots x_m^{\sigma_m}$$

- Совершенная конъюктивная нормальная форма

$$\bigwedge_{\sigma : f(\sigma)=0} x_1^{\overline{\sigma}_1} \vee \dots \vee x_m^{\overline{\sigma}_m}$$

- Полином Жегалкина

$$\bigoplus_{\sigma \in \mathbb{F}_2^m} a_\sigma x^\sigma$$

Теорема Жегалкина

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Теорема

Теорема Жегалкина: Всякая булева функция представима полиномом, и при том единственным образом.

$$f = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{23}x_2x_3 \oplus a_{123}x_1x_2x_3$$

Найдем коэффициенты многочлена следующим образом

- $f(000) = a_0$
- $f(001) = a_0 \oplus a_3 \Rightarrow a_3 = f(000) \oplus f(001) = \bigoplus_{\sigma \in 00*} f(\sigma)$
- $f(010) = a_0 \oplus a_2 \Rightarrow a_2 = \bigoplus_{\sigma \in 0*0} f(\sigma)$

Аналогично, $a_{13} = \bigoplus_{\sigma \in *0*} f(\sigma)$, $a_{123} = \bigoplus_{\sigma \in ***} f(\sigma)$

Пример получения полинома Жегалкина

$$f = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{23}x_2x_3 \oplus a_{123}x_1x_2x_3$$

x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$\begin{aligned}
a_0 &= f(000) &&= 1\\
a_1 &= \bigoplus_{\sigma\in *00} f(\sigma) &&= 1\\
a_2 &= \bigoplus_{\sigma\in 0*0} f(\sigma) &&= 0\\
a_3 &= \bigoplus_{\sigma\in 00*} f(\sigma) &&= 1\\
a_{12} &= \bigoplus_{\sigma\in **0} f(\sigma) &&= 0\\
a_{13} &= \bigoplus_{\sigma\in *0*} f(\sigma) &&= 0\\
a_{23} &= \bigoplus_{\sigma\in 0**} f(\sigma) &&= 1\\
a_{123} &= \bigoplus_{\sigma\in ***} f(\sigma) &&= 1
\end{aligned}$$

$$f = 1 \oplus x_1 \oplus x_3 \oplus x_2x_3 \oplus x_1x_2x_3$$

Коды Рида-Маллера

Определение

Множество векторов значений всевозможных булевых функций от m переменных $x_1, \ldots, x_m$ степени не больше r называется (двоичным) кодом Рида – Маллера или просто РМ – кодом.

$$RM(m, r) = \{f : \{0,1\}^m \to \{0,1\} \mid \deg f \leq r\}$$

- код $RM(m, r)$ — линейное векторное пространство над полем $\mathbb{F}_2$: т.к. $\deg(f \oplus g) \leq \max(\deg f, \deg g)$, то

$$\forall \alpha, \beta \in \mathbb{F}_2 \forall f, g \in RM(m, r) : \alpha f \oplus \beta g \in RM(m, r)$$

- длина кода $n = 2^m$
- размерность кода $k = C_0^m + C_1^m + \ldots + C_r^m$
- расстояние: $f \in RM(m, r), f \neq 0 \Rightarrow \|f\| \geq 2^{m-r}$

Конструкция Плоткина

Обозначения

- U – линейный $[n, k_U, d_U]$ код
- V – линейный $[n, k_V, d_V]$ код

Конструкция Плоткина

$$\mathcal{C} = U \triangle V = (U, U + V) = \{(u, u + v) : u \in U, v \in V\}$$

- линейный $n(\mathcal{C}) = 2n,\ k(\mathcal{C}) = k_U + k_V$ код
- кодовое расстояние $d(\mathcal{C}) = \min\{d_V, 2d_U\}$
- $RM(m, r) = RM(m - 1, r) \triangle RM(m - 1, r - 1)$

Порождающая матрица кода Рида-Маллера

$$G = \begin{bmatrix} G_0 \\ G_1 \\ \dots \\ G_r \end{bmatrix}$$

- Строки матрицы G_j – это вектора значений всех приведённых одночленов степени $= j$, одночлены лексикографически упорядочены по нижним индексам

$$RM(3,2) = \{1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3\}$$

$$G = \left[\begin{array}{c|c} 11111111 & 1 \\ \hline 00001111 & x_1 \\ 00110011 & x_2 \\ 01010101 & x_3 \\ \hline 00000011 & x_1x_2 \\ 00000101 & x_1x_3 \\ 00010001 & x_2x_3 \end{array}\right]$$

Декодирование кода Рида-Маллера

- Перебором: в ближайшее кодовое слово
- По синдрому или таблице стандартного расположения
- Мажоритарный алгоритм Рида (основан на свойствах полиномов Жегалкина)

Задача декодирования $RM(2,3)$

- Дан полином

$$f = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{23}x_2x_3$$

- Известны его значения во всех точках $\{x_1, x_2, x_3\} \in \{0,1\}^3$
- Найти коэффициенты $a_1, a_2, a_3, a_{12}, a_{13}, a_{23}$

Декодирование кода Рида-Маллера

- Перебором: в ближайшее кодовое слово
- По синдрому или таблице стандартного расположения
- Мажоритарный алгоритм Рида (основан на свойствах полиномов Жегалкина)

Задача декодирования $RM(2,3)$

- Дан полином

$$f = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{23}x_2x_3$$

- Известны его значения во всех точках $\{x_1, x_2, x_3\} \in \{0,1\}^3$
- Найти коэффициенты $a_1, a_2, a_3, a_{12}, a_{13}, a_{23}$

Идея решения

- Оценить каждый из коэффициентов $a_1, a_2, a_3, a_{12}, a_{13}, a_{23}$ несколькими независимыми способами, принять решение по мажоритарному принципу

Декодирование РМ-кода: характеристические суммы

Определение

Пусть $A \subseteq \{0,1\}^m$ – некоторое множество, $f = f(x_1, \ldots, x_m)$ – булева функция. Тогда характеристическая сумма:

$$\phi_A(f) = \bigoplus_{\sigma \in A} f(\sigma)$$

- Вес функции $\|f\|$ – это вес её вектора значений
- Пусть f_A – сужение функции f на множество A и $\|f\|_A = \|f_A\|$, тогда

$$\phi_A(f) = \begin{cases} 0, & \text{если } \|f\|_A \text{ четно} \\ 1, & \text{если } \|f\|_A \text{ нечетно} \end{cases}$$

- $\phi_{A \cup B}(f) = \phi_A(f) \oplus \phi_B(f) \oplus \phi_{A \cap B}(f)$
- $\phi_A(f \oplus g) = \phi_A(f) \oplus \phi_A(g)$

Декодирование РМ-кода: ассоциированные грани и наборы

Определение

Пусть $A \subseteq \mathbb{F}_m^2$ — грань размерности r. Набор $\sigma(A) \in \mathbb{F}_m^2$ веса r, равный нулю в разрядах, соответствующих фиксированным разрядам грани A, и равный единице в остальных разрядах, будем называть ассоциированным с гранью A.

$$\mathbf{A}(\tau) = \{A - \text{грань} : \sigma(A) = \tau\}, |\mathbf{A}(\tau)| = 2^{m-\|\tau\|}$$

Примеры

Пусть $\begin{cases} A = 1{*}0{*}{*}10 \\ B = 0{*}1{*}{*}01 \end{cases} \Rightarrow \sigma(A) = \sigma(B) = 0101100,$ *Пусть* $\tau = 10110 \Rightarrow \mathbf{A}(\tau) = \begin{matrix} {*}0{*}{*}0{*} \\ {*}0{*}{*}1{*} \\ {*}1{*}{*}0{*} \\ {*}1{*}{*}1{*} \end{matrix}$

Декодирование РМ-кода: ассоциированные конъюнкции

Определение

- *Ассоциированная с гранью A конъюнкция K:*

$$K(A) = \mathbf{x}^{\sigma(A)}$$

- *Ассоциированные с конъюнкцией K грани:*

$$\mathbf{A}(K) = \{A : K(A) = A\}, \quad K = \mathbf{x}^{\sigma} \Rightarrow \mathbf{A}(K) = \mathbf{A}(\sigma)$$

- *Нулевая грань* $A_0 \in \mathbf{A}(K) : \mathbf{0} \in A_0(K)$

Примеры

$$\textit{Пусть} \begin{cases} A = 1{*}0{*}{*}10 \\ B = 0{*}1{*}{*}01 \end{cases} \Rightarrow K(A) = K(B) = x_1^0 x_2^1 x_3^0 x_4^1 x_5^1 x_6^0 x_7^0 = x_2 x_4 x_5$$

$$K = x_2 x_4 x_5 \to A_0(K) = 0{*}0{*}{*}00$$

Декодирование РМ-кода на примере $RM(3,2)$

$$f = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{12}x_2x_3$$

- Конъюнкция x_1x_2 имеет две грани $**1$, $**0$ с характеристическим свойством:

$$a_{12} = \phi_{**1}(f) = \phi_{**0}(f), \quad \deg f \leq 2$$

- Грани $**1$, $**0$ не пересекаются, суммы $\phi_{**1}(f)$, $\phi_{**0}(f)$ вычисляются независимо
- Если инвертировать один разряд в векторе значений полинома $f, \deg f \leq 2$, то

$$\phi_{**1}(f) \neq \phi_{**0}(f)$$

- Код $RM(3,2)$ обнаруживает одну ошибку[2]
- Больше подобных проверок – больше ошибок можно обнаружить и исправить

[2] следует из свойств кодового расстояния РМ-кода

Пример: Декодирование РМ-кода на примере $RM(4,2)$

$$y = 1010 \quad 1001 \quad 1101 \quad 0110$$

$$f = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus a_3 x_3 \oplus a_4 x_4 \oplus a_{12} x_1 x_2 \oplus a_{13} x_1 x_3 \oplus a_{14} x_1 x_4 \oplus a_{23} x_2 x_3 \oplus a_{24} x_2 x_4 \oplus a_{34} x_3 x_4$$

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\mathbf{y} = \mathbf{f} + \mathbf{e}$	1	0	1	0	1	0	0	1	1	1	0	1	0	1	1	0

Пример: Декодирование РМ-кода на примере $RM(4,2)$

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\mathbf{y} = \mathbf{f} + \mathbf{e}$	1	0	1	0	1	0	0	1	1	1	0	1	0	1	1	0

A	$\phi_A(\mathbf{y})$	A	$\phi_A(\mathbf{y})$	A	$\phi_A(\mathbf{y})$	A	$\phi_A(\mathbf{y})$	A	$\phi_A(\mathbf{y})$	A	$\phi_A(\mathbf{y})$
$**00$	1	$*0*0$	1	$*00*$	1	$0**0$	1	$0*0*$	0	$00**$	0
$**01$	0	$*0*1$	0	$*01*$	0	$0**1$	1	$0*1*$	0	$01**$	0
$**10$	0	$*1*0$	0	$*10*$	0	$1**0$	0	$1*0*$	1	$10**$	1
$**11$	0	$*1*1$	0	$*11*$	0	$1**1$	1	$1*1*$	0	$11**$	0
a_{12}	0	a_{13}	0	a_{14}	0	a_{23}	1	a_{24}	0	a_{14}	0

$$f = f_1 \oplus x_2x_3, \deg f_1 \leq 1, \quad f_1 = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_4x_4$$

Пример: Декодирование РМ-кода на примере $RM(4,2)$

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\mathbf{y} = \mathbf{f} + \mathbf{e}$	1	0	1	0	1	0	0	1	1	1	0	1	0	1	1	0
$x_2 x_3$	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
$y_1 = f_1 \oplus e$	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1

A	$\phi_A(\mathbf{y})$	A	$\phi_A(\mathbf{y})$	A	$\phi_A(\mathbf{y})$	A	$\phi_A(\mathbf{y})$
$*000$	0	$0*00$	0	$00*0$	0	$000*$	1
$*001$	1	$0*01$	0	$00*1$	0	$001*$	1
$*010$	1	$0*10$	0	$01*0$	0	$010*$	1
$*011$	1	$0*11$	0	$01*1$	0	$011*$	1
$*100$	1	$1*00$	1	$10*0$	1	$100*$	0
$*101$	1	$1*01$	0	$10*1$	0	$101*$	1
$*110$	1	$1*10$	0	$11*0$	0	$110*$	1
$*111$	1	$1*11$	0	$11*1$	0	$111*$	1
a_1	1	a_2	0	a_3	0	a_4	1

Пример: Декодирование РМ-кода на примере $RM(4,2)$

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\mathbf{y} = \mathbf{f} + \mathbf{e}$	1	0	1	0	1	0	0	1	1	1	0	1	0	1	1	0
$x_2 x_3$	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
$y_1 = f_1 \oplus e$	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1
$x_1 \oplus x_4$	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
$y_0 = f_0 + e$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

- $a_0 = 1$
- $e = 0000 \quad 0000 \quad 1000 \quad 0000$

Алгоритм Рида

1. Получено слово $y = x + e$, $x \in RM(m, r)$
2. Для каждого из C_r^m одночленов x^σ степени $r \geq 0$ вычислить проверочные суммы $\phi_A(y)$ по всем $2m - r$ ассоциированным с ним граням $A = A(x^\sigma)$
3. Если нулей и единиц в множестве $\{\phi_A(y)\}_A$ поровну, то сообщить об отказе от декодирования
4. Принять коэффициент a_σ при одночлене x^σ в кодовом слове x равным наиболее часто встречающемуся числу среди сумм $\{\phi_A(y)\}$
5. Просуммировать найденные старшие члены слова x и вычесть их из слова y, будет получено слово $y' = x' + e, \quad x' \in RM(m, r-1)$
6. Повторить действия пп. 2–5 для $r = r - 1$, $y = y'$

Содержание

Полярные коды: выводы

- Конструкция предложена в 2009 году и является продолжением идеи РМ-кодов
  - В коде Рида-Маллера из проверочной матрицы удаляются строки максимального веса
  - В полярных кодах удаляются строки, соответствующие наменее надежным виртуальным каналам
- Метод последовательных исключений со списком позволил существенно улучшить эффективность декодирования
  - Наличие «оракула», подсказывающего наличие слова в списке, существенно улучшает эффективность декодера
  - Выбор одного слова из списка осуществляется с помощью CRC
  - Список длины $L = 2^k$ – MAP декодер
- Полярные коды нашли применение в стандартах сотовой связи пятого поколения

Skoltech
Skolkovo Institute of Science and Technology

Спасибо за внимание!

СИРИУС

ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР

Skoltech
Skolkovo Institute of Science and Technology