



# Códigos LDPC sobre canal binario con símbolo borrado

Pablo M. Olmos  
Manuel A. Vázquez

7 de marzo de 2024

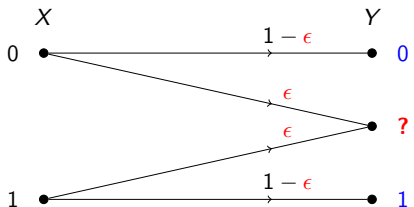
# Índice

- 1 Canal binario con símbolo borrado (BEC)
- 2 Codificación de canal clásica
- 3 Codificación de canal moderna
- 4 Códigos LDPC

# Índice

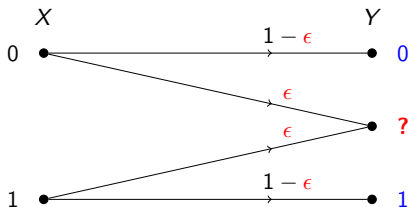
- 1 Canal binario con símbolo borrado (BEC)
- 2 Codificación de canal clásica
- 3 Codificación de canal moderna
- 4 Códigos LDPC

# El canal binario con símbolo borrado (BEC)



Capacidad:  
 $C = 1 - \epsilon$

# El canal binario con símbolo borrado (BEC)

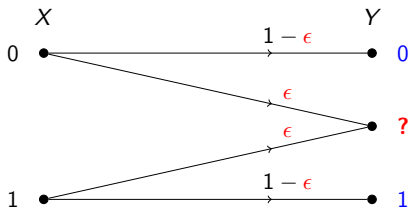


Capacidad:  
 $C = 1 - \epsilon$

El modelo es muy simple, pero aún así...

- sorprendentemente, **la mayoría de propiedades y afirmaciones que encontramos en nuestra investigación de códigos LDPC sobre BEC son mucho más generales** (R. Urbanke and T. Richardson, Modern Coding Theory) y, además,

# El canal binario con símbolo borrado (BEC)



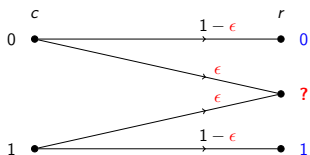
Capacidad:

$$C = 1 - \epsilon$$

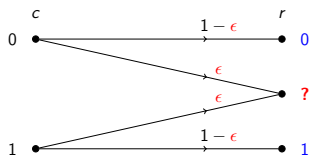
El modelo es muy simple, pero aún así...

- sorprendentemente, **la mayoría de propiedades y afirmaciones que encontramos en nuestra investigación de códigos LDPC sobre BEC son mucho más generales** (R. Urbanke and T. Richardson, Modern Coding Theory) y, además,
- en la capa de enlace de algunos sistemas de comunicaciones se usan códigos correctores de borrado.

# BEC: consideraciones prácticas



# BEC: consideraciones prácticas

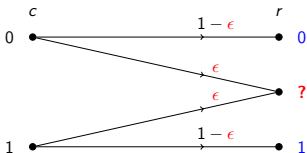


Transmisión sin codificar

Probabilidad de error de bit *del canal*  $\equiv \epsilon$



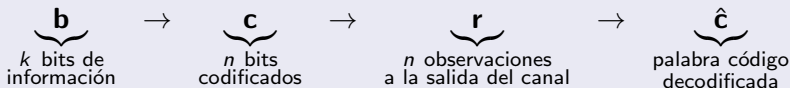
# BEC: consideraciones prácticas



## Transmisión sin codificar

Probabilidad de error de bit *del canal*  $\equiv \epsilon$

## Transmisión de bits codificados



La tasa del código sigue siendo  $R = \frac{k}{n}$

# Teorema de codificación

Podemos conseguir una probabilidad de error (de palabra código) arbitrariamente pequeña,

$$P(\hat{\mathbf{c}} \neq \mathbf{c}|\mathbf{r}) \rightarrow 0,$$

cuando  $n \rightarrow \infty$  si la tasa del código está por debajo de la capacidad, i.e.,

$$R < C.$$

# Teorema de codificación

Podemos conseguir una probabilidad de error (de palabra código) arbitrariamente pequeña,

$$P(\hat{\mathbf{c}} \neq \mathbf{c}|\mathbf{r}) \rightarrow 0,$$

cuando  $n \rightarrow \infty$  si la tasa del código está por debajo de la capacidad, i.e.,

$$R < C.$$



**No queremos esto...**

Utilizar  $n \rightarrow \infty$  es un derroche de recursos (tiempo, energía).

# Teorema de codificación

Podemos conseguir una probabilidad de error (de palabra código) arbitrariamente pequeña,

$$P(\hat{\mathbf{c}} \neq \mathbf{c}|\mathbf{r}) \rightarrow 0,$$

cuando  $n \rightarrow \infty$  si la tasa del código está por debajo de la capacidad, i.e.,

$$R < C.$$

 **No queremos esto...**

Utilizar  $n \rightarrow \infty$  es un derroche de recursos (tiempo, energía).

## Objetivo

...diseñar esquemas de codificación y decodificación **viables** que nos permitan operar cerca de la capacidad del canal.

# Índice

- 1 Canal binario con símbolo borrado (BEC)
- 2 Codificación de canal clásica
- 3 Codificación de canal moderna
- 4 Códigos LDPC

## Códigos bloque lineales

- Matriz generadora:  $\mathbf{c} = \mathbf{bG}$  donde  $\mathbf{b} \in \{0, 1\}^k$ .

## Códigos bloque lineales

- Matriz generadora:  $\mathbf{c} = \mathbf{bG}$  donde  $\mathbf{b} \in \{0, 1\}^k$ .
- Matriz de control de paridad:  $\mathbf{cH}^T = \mathbf{0} \quad \forall \mathbf{c} \in \mathcal{C}$ .
  - $\mathcal{C}$  es el conjunto de todas las palabras código (*codebook*)

## Códigos bloque lineales

- **Matriz generadora:**  $\mathbf{c} = \mathbf{bG}$  donde  $\mathbf{b} \in \{0, 1\}^k$ .
- **Matriz de control de paridad:**  $\mathbf{cH}^T = \mathbf{0} \quad \forall \mathbf{c} \in \mathcal{C}$ .
  - $\mathcal{C}$  es el conjunto de todas las palabras código (*codebook*)
- Cada fila de la matriz de control de paridad da lugar a una restricción lineal entre los bits codificados.



# Códigos bloque lineales

- **Matriz generadora:**  $\mathbf{c} = \mathbf{bG}$  donde  $\mathbf{b} \in \{0, 1\}^k$ .
- **Matriz de control de paridad:**  $\mathbf{cH}^T = \mathbf{0} \quad \forall \mathbf{c} \in \mathcal{C}$ .
  - $\mathcal{C}$  es el conjunto de todas las palabras código (*codebook*)
- Cada fila de la matriz de control de paridad da lugar a una restricción lineal entre los bits codificados.

Para un código Hamming (7, 4),

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

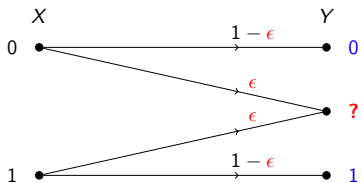
Por tanto...

$$c_1 \oplus c_3 \oplus c_5 \oplus c_7 = 0$$

$$c_2 \oplus c_3 \oplus c_6 \oplus c_7 = 0$$

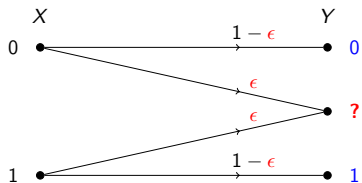
$$c_4 \oplus c_5 \oplus c_6 \oplus c_7 = 0$$

# Transmisión sobre BEC



- Código bloque lineal  $(n, k)$  con matrices  $\mathbf{G}$  y  $\mathbf{H}$ .
- Se envía la palabra código  $\mathbf{c}$ .
- Se observa el vector  $\mathbf{r}$ .

# Transmisión sobre BEC

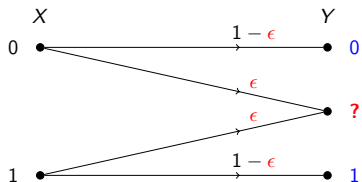


- Código bloque lineal  $(n, k)$  con matrices  $\mathbf{G}$  y  $\mathbf{H}$ .
- Se envía la palabra código  $\mathbf{c}$ .
- Se observa el vector  $\mathbf{r}$ .

El canal *borra* unos bits y otros no:

- $\mathcal{E}$  es un conjunto que contiene los índices de los bits **borrados**
- $\mathcal{R}$  es un conjunto que contiene los índices de los bits **recibidos**
- $\mathcal{E} \cup \mathcal{R} = \{1, \dots, n\}$ .

# Transmisión sobre BEC



- Código bloque lineal  $(n, k)$  con matrices  $\mathbf{G}$  y  $\mathbf{H}$ .
- Se envía la palabra código  $\mathbf{c}$ .
- Se observa el vector  $\mathbf{r}$ .

El canal *borra* unos bits y otros no:

- $\mathcal{E}$  es un conjunto que contiene los índices de los bits **borrados**
- $\mathcal{R}$  es un conjunto que contiene los índices de los bits **recibidos**
- $\mathcal{E} \cup \mathcal{R} = \{1, \dots, n\}$ .

Por tanto, para el BEC

$$\mathbf{r}(\mathcal{E}) = ?, \quad \mathbf{r}(\mathcal{R}) = \mathbf{c}(\mathcal{R})$$

# Decodificación sobre BEC: ejemplo

- Código Hamming (7, 4).
- $\mathbf{c} = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$  es enviado.
- $\mathbf{r} = [1 \ ? \ 1 \ 0 \ ? \ ? \ 0]$  es recibido.
- $\mathcal{E} = \{2, 5, 6\}$  y  $\mathcal{R} = \{1, 3, 4, 7\}$ .

Así pues, el sistema de ecuaciones se puede simplificar

$$\left. \begin{aligned} c_1 \oplus c_3 \oplus c_5 \oplus c_7 &= 0 \\ c_2 \oplus c_3 \oplus c_6 \oplus c_7 &= 0 \\ c_4 \oplus c_5 \oplus c_6 \oplus c_7 &= 0 \end{aligned} \right\}$$

# Decodificación sobre BEC: ejemplo

- Código Hamming (7, 4).
- $\mathbf{c} = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$  es enviado.
- $\mathbf{r} = [1 \ ? \ 1 \ 0 \ ? \ ? \ 0]$  es recibido.
- $\mathcal{E} = \{2, 5, 6\}$  y  $\mathcal{R} = \{1, 3, 4, 7\}$ .

Así pues, el sistema de ecuaciones se puede simplificar

$$\left. \begin{array}{l} c_1 \oplus c_3 \oplus c_5 \oplus c_7 = 0 \\ c_2 \oplus c_3 \oplus c_6 \oplus c_7 = 0 \\ c_4 \oplus c_5 \oplus c_6 \oplus c_7 = 0 \end{array} \right\} \rightarrow \left. \begin{array}{l} 1 \oplus 1 \oplus c_5 \oplus 0 = 0 \\ c_2 \oplus 1 \oplus c_6 \oplus 0 = 0 \\ 0 \oplus c_5 \oplus c_6 \oplus 0 = 0 \end{array} \right\}$$

# Decodificación sobre BEC: ejemplo

- Código Hamming (7, 4).
- $\mathbf{c} = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$  es enviado.
- $\mathbf{r} = [1 \ ? \ 1 \ 0 \ ? \ ? \ 0]$  es recibido.
- $\mathcal{E} = \{2, 5, 6\}$  y  $\mathcal{R} = \{1, 3, 4, 7\}$ .

Así pues, el sistema de ecuaciones se puede simplificar

$$\left. \begin{array}{l} c_1 \oplus c_3 \oplus c_5 \oplus c_7 = 0 \\ c_2 \oplus c_3 \oplus c_6 \oplus c_7 = 0 \\ c_4 \oplus c_5 \oplus c_6 \oplus c_7 = 0 \end{array} \right\} \rightarrow \left. \begin{array}{l} 1 \oplus 1 \oplus c_5 \oplus 0 = 0 \\ c_2 \oplus 1 \oplus c_6 \oplus 0 = 0 \\ 0 \oplus c_5 \oplus c_6 \oplus 0 = 0 \end{array} \right\} \rightarrow \left. \begin{array}{l} c_5 = 0 \\ c_2 \oplus c_6 = 1 \\ c_5 \oplus c_6 = 0 \end{array} \right\}$$

Resolviendo el sistema de ecuaciones binarias resulta una única solución  $\hat{\mathbf{c}} = [1110000] = \mathbf{c}$ .

## Decodificación sobre el BEC: formulación general

- Código bloque lineal  $(n, k)$  con matrices  $\mathbf{G}$  y  $\mathbf{H}$ .
- Se envía la palabra código  $\mathbf{c}$ .
- Se observa el vector  $\mathbf{r}$ .
- $\mathbf{H}_{\mathcal{E}}$  es la submatriz de  $\mathbf{H}$  que se obtiene eligiendo las *columnas* cuyo índices están en  $\mathcal{E}$  (y, análogamente,  $\mathbf{H}_{\mathcal{R}}$  es...).



# Decodificación sobre el BEC: formulación general

- Código bloque lineal  $(n, k)$  con matrices  $\mathbf{G}$  y  $\mathbf{H}$ .
- Se envía la palabra código  $\mathbf{c}$ .
- Se observa el vector  $\mathbf{r}$ .
- $\mathbf{H}_{\mathcal{E}}$  es la submatriz de  $\mathbf{H}$  que se obtiene eligiendo las *columnas* cuyo índices están en  $\mathcal{E}$  (y, análogamente,  $\mathbf{H}_{\mathcal{R}}$  es...).

## Decodificación óptima maximum a posteriori

Encontrar  $\mathbf{c}(\mathcal{E})$  resolviendo el siguiente sistema de ecuaciones:

$$\mathbf{c}(\mathcal{E})\mathbf{H}_{\mathcal{E}}^T = \mathbf{c}(\mathcal{R})\mathbf{H}_{\mathcal{R}}^T$$

# Decodificación sobre el BEC: formulación general

- Código bloque lineal  $(n, k)$  con matrices  $\mathbf{G}$  y  $\mathbf{H}$ .
- Se envía la palabra código  $\mathbf{c}$ .
- Se observa el vector  $\mathbf{r}$ .
- $\mathbf{H}_{\mathcal{E}}$  es la submatriz de  $\mathbf{H}$  que se obtiene eligiendo las *columnas* cuyo índices están en  $\mathcal{E}$  (y, análogamente,  $\mathbf{H}_{\mathcal{R}}$  es...).

## Decodificación óptima maximum a posteriori

Encontrar  $\mathbf{c}(\mathcal{E})$  resolviendo el siguiente sistema de ecuaciones:

$$\mathbf{c}(\mathcal{E})\mathbf{H}_{\mathcal{E}}^T = \mathbf{c}(\mathcal{R})\mathbf{H}_{\mathcal{R}}^T$$

En el ejemplo anterior:

$$\begin{bmatrix} c_2 & c_5 & c_6 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

## Sistema de ecuaciones lineales para decodificación MAP

$$\mathbf{cH}^T = [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7] \begin{bmatrix} h_1 \\ \vdots \\ h_7 \end{bmatrix} = 0$$

# Sistema de ecuaciones lineales para decodificación MAP

$$\begin{aligned} \mathbf{cH}^T &= [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7] \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_7 \end{bmatrix} = 0 \\ &= c_1\mathbf{h}_1 + c_2\mathbf{h}_2 + c_3\mathbf{h}_3 + c_4\mathbf{h}_4 + c_5\mathbf{h}_5 + c_6\mathbf{h}_6 + c_7\mathbf{h}_7 = 0 \end{aligned}$$

## Sistema de ecuaciones lineales para decodificación MAP

$$\begin{aligned} \mathbf{cH}^T &= [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7] \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_7 \end{bmatrix} = 0 \\ &= c_1 \mathbf{h}_1 + c_2 \mathbf{h}_2 + c_3 \mathbf{h}_3 + c_4 \mathbf{h}_4 + c_5 \mathbf{h}_5 + c_6 \mathbf{h}_6 + c_7 \mathbf{h}_7 = 0 \\ &= c_2 \mathbf{h}_2 + c_5 \mathbf{h}_5 + c_6 \mathbf{h}_6 + c_1 \mathbf{h}_1 + c_3 \mathbf{h}_3 + c_4 \mathbf{h}_4 + c_7 \mathbf{h}_7 = 0 \end{aligned}$$

## Sistema de ecuaciones lineales para decodificación MAP

$$\begin{aligned}
 \mathbf{cH}^T &= [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7] \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_7 \end{bmatrix} = 0 \\
 &= c_1 \mathbf{h}_1 + c_2 \mathbf{h}_2 + c_3 \mathbf{h}_3 + c_4 \mathbf{h}_4 + c_5 \mathbf{h}_5 + c_6 \mathbf{h}_6 + c_7 \mathbf{h}_7 = 0 \\
 &= c_2 \mathbf{h}_2 + c_5 \mathbf{h}_5 + c_6 \mathbf{h}_6 + c_1 \mathbf{h}_1 + c_3 \mathbf{h}_3 + c_4 \mathbf{h}_4 + c_7 \mathbf{h}_7 = 0 \\
 &= [c_2 \quad c_5 \quad c_6] \underbrace{\begin{bmatrix} \mathbf{h}_2 \\ \mathbf{h}_5 \\ \mathbf{h}_6 \end{bmatrix}}_{\mathbf{H}_{\mathcal{E}}^T} + [c_1 \quad c_3 \quad c_4 \quad c_7] \underbrace{\begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \\ \mathbf{h}_7 \end{bmatrix}}_{\mathbf{H}_{\mathcal{R}}^T} = 0
 \end{aligned}$$

## Sistema de ecuaciones lineales para decodificación MAP

$$\begin{aligned}
 \mathbf{c}\mathbf{H}^T &= [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7] \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_7 \end{bmatrix} = 0 \\
 &= c_1\mathbf{h}_1 + c_2\mathbf{h}_2 + c_3\mathbf{h}_3 + c_4\mathbf{h}_4 + c_5\mathbf{h}_5 + c_6\mathbf{h}_6 + c_7\mathbf{h}_7 = 0 \\
 &= c_2\mathbf{h}_2 + c_5\mathbf{h}_5 + c_6\mathbf{h}_6 + c_1\mathbf{h}_1 + c_3\mathbf{h}_3 + c_4\mathbf{h}_4 + c_7\mathbf{h}_7 = 0 \\
 &= [c_2 \quad c_5 \quad c_6] \underbrace{\begin{bmatrix} \mathbf{h}_2 \\ \mathbf{h}_5 \\ \mathbf{h}_6 \end{bmatrix}}_{\mathbf{H}_{\mathcal{E}}^T} + [c_1 \quad c_3 \quad c_4 \quad c_7] \underbrace{\begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \\ \mathbf{h}_7 \end{bmatrix}}_{\mathbf{H}_{\mathcal{R}}^T} = 0
 \end{aligned}$$

Así pues,

$$[c_2 \quad c_5 \quad c_6] \mathbf{H}_{\mathcal{E}}^T = [c_1 \quad c_3 \quad c_4 \quad c_7] \mathbf{H}_{\mathcal{R}}^T$$

$\mathbf{h}_j \equiv j$ -ésima fila de  $\mathbf{H}^T = j$ -ésima columna de  $\mathbf{H}$

## Decodificación MAP óptima (aproximación clásica)

Al resolver el sistema de ecuaciones lineales,  $\mathbf{c}(\mathcal{E})\mathbf{H}_{\mathcal{E}}^T = \mathbf{c}(\mathcal{R})\mathbf{H}_{\mathcal{R}}^T$   
para encontrar  $\mathbf{c}(\mathcal{E})$ , hay dos posibles resultados



## Decodificación MAP óptima (aproximación clásica)

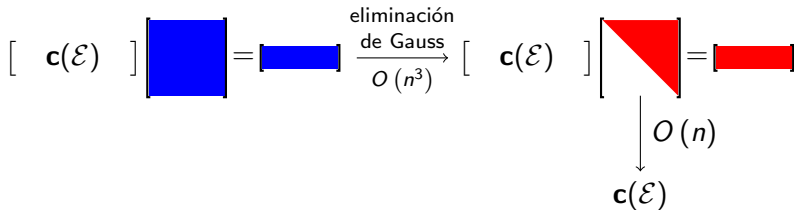
Al resolver el sistema de ecuaciones lineales,  $\mathbf{c}(\mathcal{E})\mathbf{H}_{\mathcal{E}}^T = \mathbf{c}(\mathcal{R})\mathbf{H}_{\mathcal{R}}^T$  para encontrar  $\mathbf{c}(\mathcal{E})$ , hay dos posibles resultados

- el sistema tiene múltiples soluciones  $\rightarrow$  todas son igualmente probables y decimos que ha ocurrido un **error de decodificación**.

# Decodificación MAP óptima (aproximación clásica)

Al resolver el sistema de ecuaciones lineales,  $\mathbf{c}(\mathcal{E})\mathbf{H}_{\mathcal{E}}^T = \mathbf{c}(\mathcal{R})\mathbf{H}_{\mathcal{R}}^T$  para encontrar  $\mathbf{c}(\mathcal{E})$ , hay dos posibles resultados

- el sistema tiene múltiples soluciones  $\rightarrow$  todas son igualmente probables y decimos que ha ocurrido un **error de decodificación**.
- el sistema tiene una única solución  $\rightarrow \hat{\mathbf{c}} = \mathbf{c}$ , y **no hay error de decodificación**.



Complejidad computacional:

- Eliminación de Gauss requiere  $O(n^3)$  operaciones
- Después de eliminación, sustitución hacia atrás es  $O(n)$

# Índice

- 1 Canal binario con símbolo borrado (BEC)
- 2 Codificación de canal clásica
- 3 Codificación de canal moderna**
- 4 Códigos LDPC

# Decodificación subóptima sobre BEC: ejemplo I

Sea:

- Código Hamming (7, 4)
- $\mathbf{c} = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$  es enviada.
- $\mathbf{r} = [1 \ ? \ 1 \ 0 \ ? \ ? \ 0]$  es recibida.

# Decodificación subóptima sobre BEC: ejemplo I

Sea:

- Código Hamming (7, 4)
- $\mathbf{c} = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$  es enviada.
- $\mathbf{r} = [1 \ ? \ 1 \ 0 \ ? \ ? \ 0]$  es recibida.

Suponiendo que el sistema ya está triangularizado y revela toda la información...

$$\begin{array}{l}
 \curvearrowright c_5 = 0 \\
 \curvearrowright c_5 + c_6 = 0 \rightarrow c_6 = 0 \\
 \curvearrowright c_2 + c_6 = 1 \rightarrow c_2 = 1
 \end{array}$$

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 \\ c_6 \\ c_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$



## Decodificación

La complejidad es  $O(n)$ .

# Decodificación subóptima sobre BEC: ejemplo II

Otra transmisión:

- Código Hamming (7, 4)
- $\mathbf{c} = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$  es enviada.
- $\mathbf{r} = [0 \ 1 \ ? \ 0 \ 0 \ ? \ ?]$  es recibida.

# Decodificación subóptima sobre BEC: ejemplo II

Otra transmisión:

- Código Hamming (7, 4)
- $\mathbf{c} = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$  es enviada.
- $\mathbf{r} = [0 \ 1 \ ? \ 0 \ 0 \ ? \ ?]$  es recibida.

Ahora,

$$c_3 \oplus c_7 = 0$$

$$c_3 \oplus c_6 \oplus c_7 = 1$$

$$c_6 \oplus c_7 = 0$$



## Error de decodificación

No hay ecuaciones con una sola variable. No se puede revelar ninguna información.

(si utilizáramos decodificación **óptima**,  $c_3$  se obtiene ( $c_3 = 1$ ) sumando las dos últimas ecuaciones)

# Teoría de códigos moderna vs clásica

## Clásica

- Decodificación **óptima** via ML/MAP con  $O(n^3)$  operaciones. Restringe los códigos que se pueden utilizar en la práctica.
- **Longitud (n) pequeña** porque de lo contrario la complejidad de decodificación es prohibitiva. **No podemos operar cerca de la capacidad** con probabilidad de error arbitrariamente pequeña.
- Ejemplos: Códigos bloque lineales (BCH, Reed Solomon), códigos convolucionales...

## Moderna

- Decodificación **aproximada** con peor rendimiento pero mucha menor complejidad computacional ( $O(n)$  operaciones).
- Se consigue operar **cerca de la capacidad** con probabilidad de error arbitrariamente pequeña utilizando **códigos muy largos!** (n grande)
- Ejemplos: Códigos turbo, **LDPC**, polares.



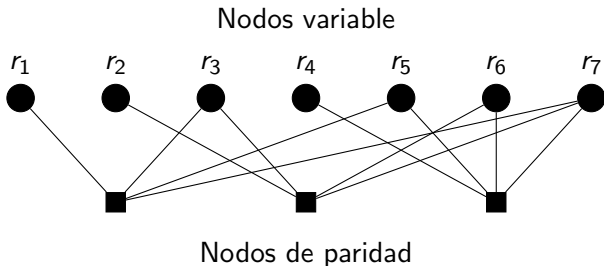
# Grafo de Tanner

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

# Grafo de Tanner

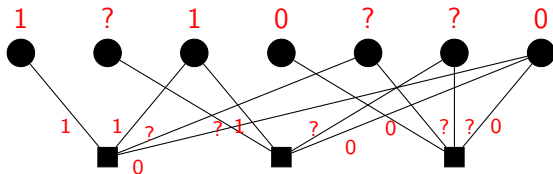
$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Las restricciones dadas por esta matriz se pueden representar utilizando un **grafo de Tanner**



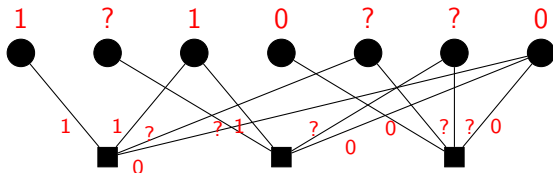
# Algoritmo de propagación de creencias (belief propagation)

**Inicialización:** los nodos variable envían las observaciones del canal a los nodos de paridad a los que están conectados:



# Algoritmo de propagación de creencias (belief propagation)

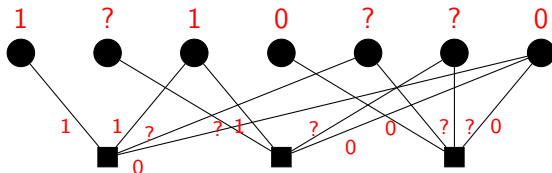
**Inicialización:** los nodos variable envían las observaciones del canal a los nodos de paridad a los que están conectados:



Mientras haya "?" no resueltos

# Algoritmo de propagación de creencias (belief propagation)

**Inicialización:** los nodos variable envían las observaciones del canal a los nodos de paridad a los que están conectados:

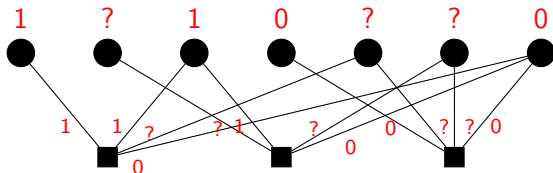


**Mientras** haya "?" no resueltos

- 1 Utilizando la información recibida, cada *nodo de paridad* intenta resolver el valor de la variable que ha enviado un mensaje "?". Si es posible, envían el valor obtenido a los nodos variable conectados. Si no, envían "?".
  - **Sólo los nodos de paridad con una única incógnita pueden hallar el valor de dicha variable!**

# Algoritmo de propagación de creencias (belief propagation)

**Inicialización:** los nodos variable envían las observaciones del canal a los nodos de paridad a los que están conectados:

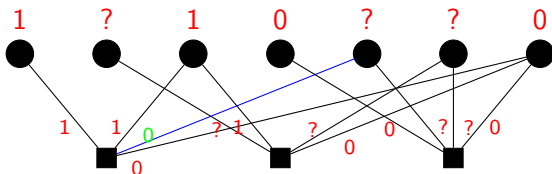


**Mientras** haya "?" no resueltos

- 1 Utilizando la información recibida, cada *nodo de paridad* intenta resolver el valor de la variable que ha enviado un mensaje "?". Si es posible, envían el valor obtenido a los nodos variable conectados. Si no, envían "?".
  - **Sólo los nodos de paridad con una única incógnita pueden hallar el valor de dicha variable!**
- 2 Los *nodos variable* envían el nuevo valor a los nodos de paridad...o reenvían el mensaje "?".

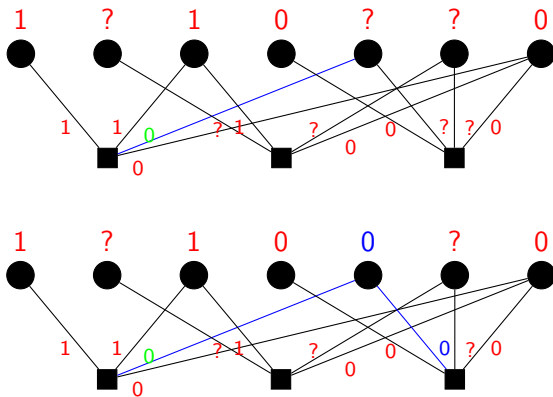
# Algoritmo de propagación de creencias (belief propagation)

## Primera iteración



# Algoritmo de propagación de creencias (belief propagation)

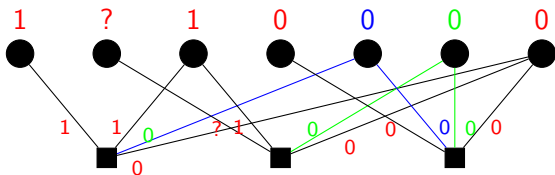
## Primera iteración





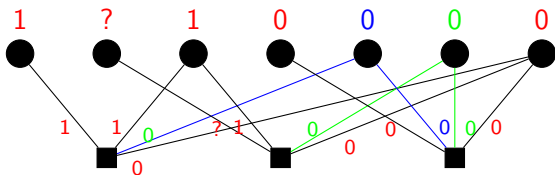
# Algoritmo de propagación de creencias (belief propagation)

Segunda iteración

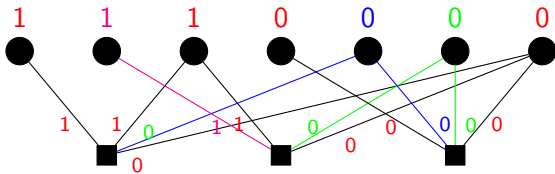


# Algoritmo de propagación de creencias (belief propagation)

## Segunda iteración



## Tercera iteración



# Algoritmo de propagación de creencias (belief propagation)

Algunas observaciones:

- En general, el rendimiento obtenido con el decodificador subóptimo es bastante malo (muchos errores de decodificación).

# Algoritmo de propagación de creencias (belief propagation)

Algunas observaciones:

- En general, el rendimiento obtenido con el decodificador subóptimo es bastante malo (muchos errores de decodificación).
- Dada una matrix de control de paridad,  $\mathbf{H}$ , de dimensiones  $(n - k) \times n$ , el número de unos por fila puede llegar a ser  $n$ .

# Algoritmo de propagación de creencias (belief propagation)

Algunas observaciones:

- En general, el rendimiento obtenido con el decodificador subóptimo es bastante malo (muchos errores de decodificación).
- Dada una matrix de control de paridad,  $\mathbf{H}$ , de dimensiones  $(n - k) \times n$ , el número de unos por fila puede llegar a ser  $n$ .
- Si una fila tiene  $\alpha n$  unos, entonces la probabilidad de que se reciban correctamente  $\alpha n - 1$  variables y se borre **exactamente una** es

$$\alpha n \epsilon (1 - \epsilon)^{(\alpha n - 1)}$$

que tiende a 0 ( $\Rightarrow$  error de decodificación) cuando  $n \rightarrow \infty$ .

( $\alpha \in (0, 1) \equiv$  proporción de 1s por bit)

# Índice

- 1 Canal binario con símbolo borrado (BEC)
- 2 Codificación de canal clásica
- 3 Codificación de canal moderna
- 4 Códigos LDPC**

# Códigos LDPC

Códigos LDPC: códigos bloque lineales definidos por una matriz de control de paridad dispersa (*sparse*)

$$\mathbf{H}_{(n-k) \times n}, \quad \mathbf{c}\mathbf{H}^T = \mathbf{0} \quad \forall \mathbf{c} \in \mathcal{C}$$

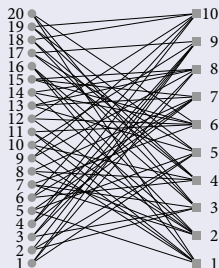
# Códigos LDPC

Códigos LDPC: códigos bloque lineales definidos por una matriz de control de paridad dispersa (*sparse*)

$$\mathbf{H}_{(n-k) \times n}, \quad \mathbf{c}\mathbf{H}^T = \mathbf{0} \quad \forall \mathbf{c} \in \mathcal{C}$$

LDPC (3,6) con  $n = 20$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0
4	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	1	1	0
5	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1
6	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	0	0	1
7	0	0	0	1	1	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0
8	1	0	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0
9	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
10	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	1



La densidad de unos en  $\mathbf{H}$  es  $6/n$  y la tasa del código  $R = 0,5$ .



## Forzando una estructura en $\mathbf{H}$

Si el número de unos por fila es siempre 6, e.g.,

$$[c_1 \quad c_2 \quad \cdots \quad c_{20}] \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} = 0 \Rightarrow c_5 + c_9 + c_{10} + c_{11} + c_{16} + c_{20} = 0$$

## Forzando una estructura en $\mathbf{H}$

Si el número de unos por fila es siempre 6, e.g.,

$$[c_1 \quad c_2 \quad \cdots \quad c_{20}] \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} = 0 \Rightarrow c_5 + c_9 + c_{10} + c_{11} + c_{16} + c_{20} = 0$$

...entonces la probabilidad de que una fila en  $\mathbf{H}$  de lugar a una ecuación con una única incógnita, e.g.,

$$c_5 + ? + c_{10} + c_{11} + c_{16} + c_{20} = 0$$

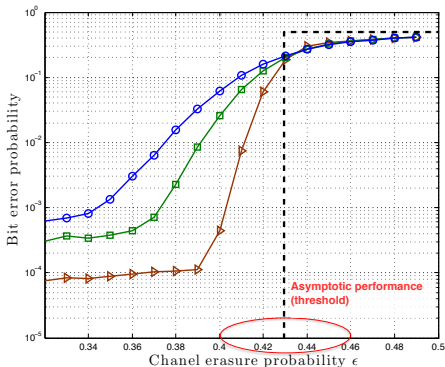
...es

$$6\epsilon(1 - \epsilon)^5,$$

ya **no depende** de  $n$ . Una ecuación con una sola incógnita se puede resolver...y una vez conocida dicha variable, hay una **probabilidad no nula de que aparezca otra ecuación en la que sólo hay una incógnita**. Esta probabilidad tampoco depende de  $n$ .

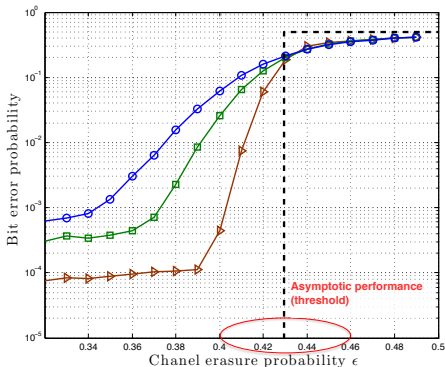
## BER sobre BEC

Tasa de error de bit del código LDPC (3, 6) sobre el BEC;  $n = 2^8$   
(○),  $n = 2^9$  (□),  $n = 2^{11}$  (▷).



# BER sobre BEC

Tasa de error de bit del código LDPC (3,6) sobre el BEC;  $n = 2^8$  ( $\circ$ ),  $n = 2^9$  ( $\square$ ),  $n = 2^{11}$  ( $\triangleright$ ).



Se puede calcular el umbral,  $\epsilon^*$ , analíticamente. Únicamente depende del **patrón de conectividad en la matrix H!**