

Guion de ayuda para la Práctica 05

Simulación de una Codificación Binaria Lineal Con Ruido

Como podéis leer en el enunciado de la Práctica 05, los procesos descritos tanto para la codificación de la fuente como para la codificación lineal son los que ya usamos en la práctica anterior. Sin embargo, ahora se añade una dificultad adicional. En este caso, se ha aplicado un simulador de ruido a la secuencia binaria que sale tras hacer la codificación lineal, que es la que se envía por el canal.

Simulador de ruido. La matriz generadora de la codificación lineal, la matriz G , tiene 15 columnas.

$$G = \left(\begin{array}{cccccc|cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right).$$

Luego las palabras código que se transmiten por el canal tienen longitud 15. Como os indico en el enunciado de esta práctica, el código lineal tiene distancia de Hamming 5. Por otra parte, la capacidad correctora del código depende de d , su distancia de Hamming, y se calcula como

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

que en este caso resulta ser 2. Esto significa que el ruido que podemos corregir de manera fiable sólo puede afectar, como mucho, a dos de las 15 posiciones de cada palabra código. Teniendo en cuenta estos aspectos teóricos, la simulación del ruido la he implementado cumpliendo el requisito anterior. De esta forma, el criterio de mínima distancia de Hamming aplicado a la lista L , que recibimos del canal, nos permite corregir el ruido.

Implementación del corrector. Para resolver la práctica, es decir, obtener el mensaje original, lo primero que tenéis que hacer es programar el proceso de corrección del ruido.

Para dicha corrección hay que construir el **tablero de síndromes incompleto**. La función *tablero_sindromes*, que tenéis en el archivo *plantilla_practica_05*, realiza dicha construcción. Su implementación está basada en los siguientes aspectos teóricos de los códigos lineales:

1. Para cualquier código lineal q -ario, con longitud n y capacidad correctora t , los errores patrón del tablero incompleto son todas las palabras q -arias de longitud n y peso menor o igual que t . En nuestro caso, son todas las palabras binarias de longitud 15 y peso menor o igual que 2. Si hacéis la combinatoria resultan ser 121 palabras.
2. Para cada error patrón, su síndrome se calcula como el producto de la matriz de control, H , por el error patrón. Cuando la matriz generadora es estándar, $G = (I_k|A)$, el cálculo de H se puede hacer con la construcción $H = (-A^T|I_{n-k})$. En nuestro caso, caso binario,

todas las cuentas se hacen en el cuerpo finito con dos elementos, que lo identificamos con \mathbb{Z}_2 (cuentas con enteros módulo dos). La matriz de control es $H = (-A^T | I_9)$ (el símbolo menos lo podéis omitir al estar trabajando módulo dos), y los síndromes obtenidos serán palabras binarias de longitud 9.

Una vez calculado el tablero de síndromes incompleto, para corregir tendréis que usar el **algoritmo del líder**. Es decir:

- Para cada palabra “recibida del canal” que tenga longitud 15 (en clase de teoría la denotamos \underline{y}), calculáis su síndrome \underline{s} .
- Buscáis en el tablero de síndromes incompleto (construido previamente) el error patrón que tenga síndrome \underline{s} .
- Si el error patrón obtenido es \underline{e} , para corregir la secuencia \underline{y} hay que hacer la cuenta $\underline{y} - \underline{e}$ módulo 2 (en general, módulo p). La secuencia obtenida con la cuenta anterior ya es fiable, es decir, seguro que no tiene ruido.

El proceso de corrección indicado en el algoritmo del líder hay que hacerlo para cada secuencia “recibida del canal” que tenga longitud n . En nuestro caso, habría que dividir la secuencia indicada en L en palabras de longitud 15, y aplicar a cada una de ellas el proceso de corrección indicado en el algoritmo del líder. Si calculáis la longitud de L resulta ser 3230. Al hacer la división euclídea de 3230 entre 15 obtenemos que el cociente es 215 y el resto es 5. Es decir, para estos datos, el proceso de corrección hay que aplicárselo a 215 secuencias de longitud 15, y nos queda una cola con 5 símbolos, que no necesita ser corregida (en la simulación del ruido no se han modificado los símbolos de la cola).

Una vez aplicado el corrector a la lista L (lista del fichero *entrada_datos_05*), el resultado ya es una secuencia fiable, y para obtener el mensaje original tendremos que usar los procesos de decodificación lineal y de decodificación de la fuente de la práctica anterior.