Teoría de la Información y la Codificación

Ejercicios – Tema 4

Curso 2017/2018

Francisco Javier Caracuel Beltrán

caracuel@correo.ugr.es

4º - Grado en Ingeniería Informática – CCIA – ETSIIT

Índice

1. ¿Qué es la redundancia?

Es el proceso de transmisión de información que implica añadir términos que refuerzan otros ya enviados.

2. ¿Qué tipos de errores conoces? Explica en qué consiste cada tipo de error. Expón un ejemplo de cada uno.

- Aislados: los errores no están seguidos (juntos) en la trama. Pueden ser:
  - Simples: en la trama se tiene un error.
    Ejemplo:
    - Enviado: 00110011
    - Recibido: 01110011
  - Múltiples: en la trama hay múltiples errores, pero no están seguidos.
    Ejemplo:
    - Enviado: 00110011
    - Recibido: 10110111
- Ráfagas: los errores están seguidos (juntos) en la trama.
  Ejemplo:
  - Enviado: 00110011
  - Recibido: 11000011

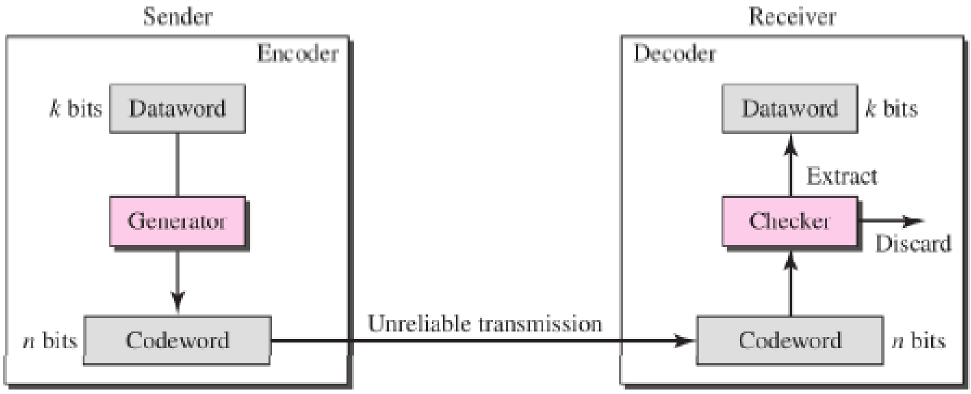
3. ¿Qué es un código de bloque? Explica la diferencia entre un código de bloque y un código uniforme, apoyándose de un ejemplo.

Un código de bloque es un código que envía el mensaje en bloques de n bits. Estos n bits incluyen el mensaje junto con un código de redundancia.

Un código de bloque es un código uniforme al tener todos los bloques la misma cantidad de símbolos, pero un código de bloque se compone de un código uniforme con k bits, al que se le añaden r bits de redundancia.

- Código uniforme:
  - A: 0000; B: 0001; C: 0010
  - A: 0000**00**; B: 0001**01**; C: 0010**10**

4. **Dibuja el diagrama del modelo de un sistema para detección de errores y, explica cada una de sus componentes.**



- Dataword: contiene la palabra que se quiere enviar con k bits o la palabra que obtiene el receptor.
- Generator: generador que se encarga de añadir redundancia a la palabra que se envía.
- Codeword: código que ha creado el generador de n bits con k bits de la palabra original y r bits de la redundancia que recibe el receptor.
- Checker: comprueba si el código recibido es válido utilizando los bits redundantes para esa validación. En caso de ser válido, guarda la palabra original y se deshace de los bits de redundancia.

5. **¿Qué es una matriz de codificación? ¿Para qué se utiliza? Expón un ejemplo de matriz de codificación para un canal con ruido, distinta de los ejemplos expuestos en clase.**

Es una matriz que representa las probabilidades de recibir un símbolo y su asociación con el símbolo enviado.

Las filas representan los caracteres del alfabeto de la fuente y las columnas representan los caracteres del alfabeto del receptor. El número de filas y columnas pueden ser diferentes porque los alfabetos pueden ser distintos.

La matriz de codificación es una tabla de probabilidad condicionada.

Se utiliza para saber la probabilidad de recibir un símbolo x habiendo enviado un símbolo y.

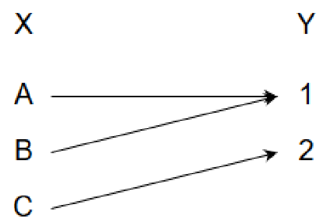
Ejemplo:

	A	B	C
0	0.80	0.05	0.15
1	0.75	0.1	0.15
2	0.60	0.2	0.2
3	0.9	0.05	0.05

6. ¿Qué es un canal determinista? Expón un ejemplo de canal de este tipo, distinto de los expuestos en clase.

Es un canal que, sabiendo lo que envía la fuente, se sabe lo que recibe el destino. Sabiendo lo que se recibe no se puede saber lo que se ha enviado.

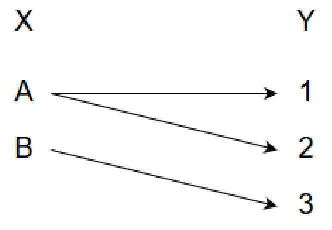
Ejemplo:



7. ¿Qué es un canal sin pérdida? Expón un ejemplo de canal de este tipo, distinto de los expuestos en clase.

Es un canal que, sabiendo lo que se recibe en el destino, se sabe qué es lo que se ha enviado.

Ejemplo:

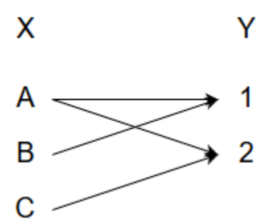


8. **¿En qué se diferencian los canales deterministas de los canales sin pérdida? Ayúdate de los ejemplos expuestos en los ejercicios anteriores para dar su respuesta.**

Un canal determinista es lo opuesto a un canal sin pérdida. En el canal determinista se sabe que recibe el destino si se sabe que envía la fuente, pero en el canal sin pérdida no. En el canal sin pérdida se sabe qué ha enviado el emisor si se sabe lo que se recibe, pero en un determinista no.

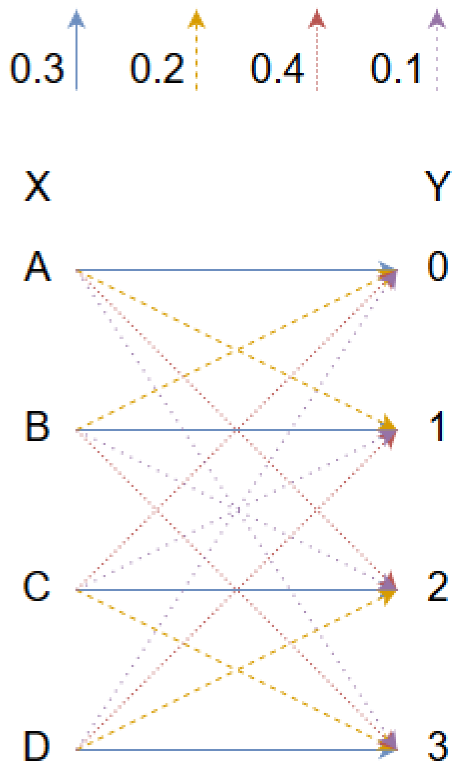
9. **¿Qué es un canal sin ruido? Expón un ejemplo de canal de este tipo, distinto de los expuestos en clase.**

Es un canal determinista y sin pérdida. Es una aplicación no biyectiva del emisor y receptor.



10. ¿Qué es un canal simétrico? Expón un ejemplo del diagrama de un canal simétrico donde tanto la fuente como el destino tienen un alfabeto de 4 símbolos. Escribe también la matriz de codificación. Indica las suposiciones realizadas para construir dicha matriz, si has debido de tomar alguna.

Es un canal donde su matriz de codificación tiene todas las filas o todas las columnas con los mismos valores en distinto orden. Si se tienen los conjuntos A_1, A_2, …, A_n, siendo *n* el número de filas o columnas de la matriz y A_i el conjunto de valores que componen la fila o columna i, la diferencia entre todos los conjuntos es el conjunto vacío.

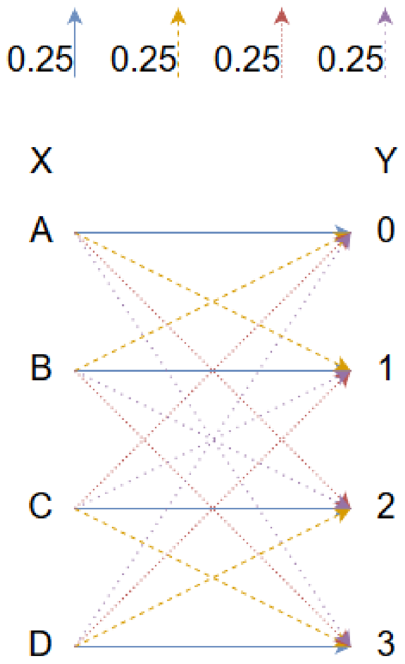


	0	1	2	3
0	0.3	0.2	0.4	0.1
1	0.2	0.3	0.1	0.4
2	0.4	0.1	0.3	0.2
3	0.1	0.4	0.2	0.3

Se han supuesto cuatro probabilidades distintas y se han intercambiado su valor entre las distintas filas y columnas de la matriz de codificación.

11. ¿Qué es un canal inútil? Expón un ejemplo del diagrama de un canal de este tipo donde tanto la fuente como el destino tienen un alfabeto de 4 símbolos. Escribe también la matriz de codificación. Indica las suposiciones realizadas para construir dicha matriz, si has debido de tomar alguna.

Es un canal que no se puede utilizar para transmitir nada. Existe la misma probabilidad de recibir cualquier símbolo enviando cualquier otro. Aporta la máxima información. Estos canales son tan ruidosos que no son practicables.



	0	1	2	3
0	0.25	0.25	0.25	0.25
1	0.25	0.25	0.25	0.25
2	0.25	0.25	0.25	0.25
3	0.25	0.25	0.25	0.25

Las suposiciones tomadas son que todos los símbolos tienen la misma probabilidad de ser recibidos por el receptor.

12. ¿Qué es la distancia de Hamming? Expón un ejemplo.

Es una distancia que mide la diferencia entre dos códigos discretos. Calcula la cantidad de puntos que no coinciden entre dos elementos.

Ejemplo:

- 000101000.
- 000010000.
- Distancia de Hamming: 3.

13. ¿Qué es la distancia de un código? Expón un ejemplo con un código inventado con k=5 palabras, donde cada palabra tenga n=8 bits.

Es la mínima distancia de Hamming que puede existir entre dos palabras, eligiendo todas las posibles palabras de un código.

Símbolo	Código
A	10011001
B	00110100
C	11001111
D	00011111
E	11111111

- D(10011001, 00110100) = 5
- D(10011001, 11001111) = 4
- D(10011001, 00011111) = 3
- D(10011001, 11111111) = 4
- D(00110100, 11001111) = 7
- D(00110100, 00011111) = 4
- D(00110100, 11111111) = 5
- D(11001111, 00011111) = 3
- D(11001111, 11111111) = 2
- D(00011111, 11111111) = 3

La distancia del código anterior es 2.

14. **Describe el Teorema de Hamming para detección de errores. Para el código expuesto en el ejercicio anterior, indica cuántos errores de un bit se asegura poder detectar en una palabra mal recibida.**

Siendo la distancia de un código *d*, el Teorema de Hamming para detección de errores indica que es posible detectar d-1 errores de 1 bit en una palabra de ese código.

Si la distancia de un código es 1, no se puede asegurar detectar algún error, porque el valor es 0.

No se puede asegurar, tampoco, que se detecten errores en un número mayor de bits que el valor obtenido.

Teniendo en cuenta el Teorema de Hamming, para el código anterior se asegura detectar 1 error.

15. **Basándose en el Teorema de Hamming para detección de errores, diseña un código binario que permita detectar 3 errores y que tenga k=8 palabras.**

Símbolo	Código
A	00000000
B	00001111
C	11110000
D	11111111
E	01010101
F	10101010
G	11001100
H	00110011

La distancia del código es 4, por lo que se puede asegurar que detecte 3 errores.

16. Basándose en el Teorema de Hamming para detección de errores, diseña un código ternario que permita detectar 3 errores y que tenga k=8 palabras. Utiliza los dígitos 0, 1 y 2 para describir los trits del código.

Símbolo	Código
A	00020000
B	00001111
C	12110000
D	11111111
E	01010101
F	10101020
G	11001100
H	00110011

La distancia del código es 4, por lo que se puede asegurar que detecte 3 errores.

17. Supón la siguiente matriz de codificación entre una fuente F (filas) con alfabeto {A, B, C} y un destino D (columnas) con alfabeto {1, 2}:

	1	2
A	0.1	0.9
B	0.9	0.1
C	0.9	0.1

Asumiendo que la fuente emite A con probabilidad p(F=A) = 0.5, que emite B con probabilidad p(F=B) = 0.3 y, que emite C con probabilidad p(F=C) = 0.2, responde a las siguientes cuestiones:

1. ¿Cuál es la probabilidad de que D perciba un 1? ¿Y de que perciba un 2?

$$p('1') = 0.1 * 0.5 + 0.9 * 0.3 + 0.9 * 0.2 = 0.5$$

$$p('2') = 0.9 * 0.5 + 0.1 * 0.3 + 0.1 * 0.2 = 0.5$$

2. ¿Cuál es la entropía de la fuente? ¿y la del destino?

$$\boldsymbol{H(S)} = -(0.5 * \log_2(0.5) + 0.5 * \log_2(0.5)) = 1$$

$$H(D|F =' A') = -(0.1 * \log_2(0.1) + 0.9 * \log_2(0.9)) = 0.469$$

$$H(D|F =' B) = -(0.9 * \log_2(0.9) + 0.1 * \log_2(0.1)) = 0.469$$

$$H(D|F =' C') = -(0.9 * \log_2(0.9) + 0.1 * \log_2(0.1)) = 0.469$$

$$H(D|F) = 0.5 * 0.469 + 0.3 * 0.469 + 0.2 * 0.469 = 0.469$$

$$\boldsymbol{H(D)} = 0.469 * 3 = 1.407$$

3. ¿Cuál es la información mutua?

$$I(D; F) = 1 - 0.469 = 0.531$$

18. **Supongamos un canal binario simétrico por el que se pueden transmitir bits con una probabilidad de error de p=0.25. ¿Cuál sería la capacidad del canal? ¿Cuántas unidades de tiempo necesitaríamos para enviar 1 bit de información?**

$C(0.25) = 1 + 0.25 * \log_2(0.25) + (1 - 0.25) * \log_2(1 - 0.25) = 0.18872$ es la capacidad del canal.

1 / 0.18872 = 5.299. Para enviar 1 bit de información son necesarias 6 unidades de tiempo.

19. **Enuncia el Teorema de Shannon para codificación con ruido.**

Si existe un canal simétrico binario con probabilidad de error p y existe un número R que satisface $0<R<C(p)$, entonces, existe para cualquier $\varepsilon>0$ y un *n* lo suficientemente grande, un $(n, 2^k, d)$-código con ratio $k/n<=R$ tal que $e(C)< \varepsilon$.

Si el canal es binario y simétrico, tiene ruido con probabilidad *p* y no es un canal inútil, entonces existe un código que se puede transmitir por el canal, debiendo tener ese código menos o igual 2^k palabras de longitud n y, $k/n < C(p)$.

20. **Suponiendo que codificamos un código uniforme con 4 símbolos (00, 01, 10, 11), ¿sería posible transmitir este código tal cual por el canal del ejercicio 18, según el Teorema de Shannon y, poder recuperar la información en el destino?**

21. **Genera un código binario uniforme para codificar un alfabeto de la fuente de 4 símbolos, con palabras de longitud *n* (a escoger) que, según el Teorema de Shannon, pueda utilizarse para recibir el mensaje en el destino correctamente, en el escenario del ejercicio 18 y, que asegure detectar 2 errores.**

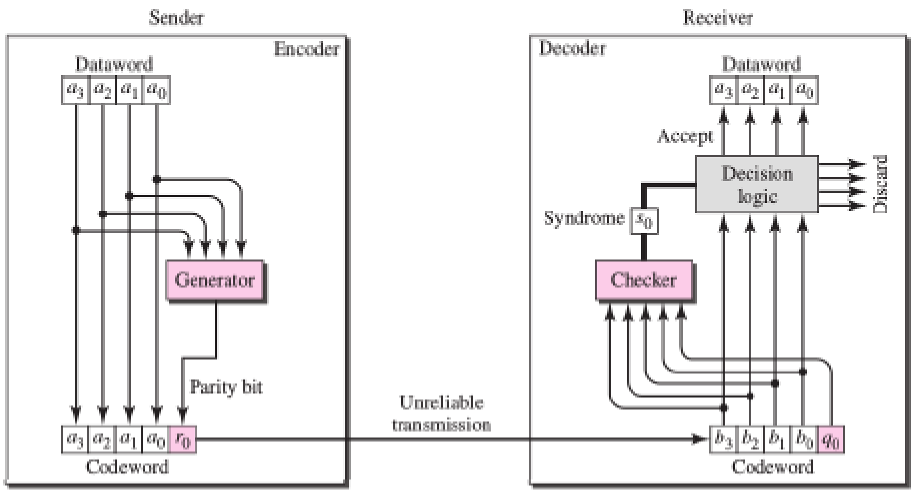
22. **¿Qué son los bits de paridad? ¿Para qué se utilizan? ¿Cuál es la diferencia entre paridad par y paridad impar? Dibuja el esquema para detección de errores con bits de paridad, explicando sus componentes.**

Son unos bits que se añaden al código y permiten, contando el número de *0s* y *1s*, comprobar la paridad de dicho código.

Los bits de paridad se utilizan para detectar errores simples o de ocurrencia impar.

La paridad par indica que la suma de todos los bits es 0.

La paridad impar indica que la suma de todos los bits es 1.



- Existe un emisor que comprueba la paridad del código. Dependiendo de si se quiere paridad par o impar, se añade un último bit al final.
- El receptor recibe el código y comprueba la paridad del código. Dependiendo de la paridad, acepta o rechaza el código.

23. Exponga un ejemplo de bits de paridad para codificar la palabra (00110101), indicando qué valor debería tomar el bit de paridad en caso de usar paridad par y cuál valor en caso de utilizar paridad impar.

- Paridad par: **0**0110101.
- Paridad impar: **1**0110101.

24. ¿En qué consiste la paridad bidimensional HVC? ¿Qué es un código P(m, k)? Expón un ejemplo de codificación HVC con m=2 para transmitir el mensaje 11010110.

Consiste en agrupar bits de paridad, agrupando el mensaje a enviar por bloques y aplicar bits de paridad horizontal, vertical y cruzada.

Si se entiende un mensaje a enviar como una agrupación de bits en un bloque, un código P(m,k) es una matriz m, k donde se introduce dicho código. Se debe añadir una fila y una columna para los bits de paridad

1	**1**	**0**	**1**	1
0	**1**	**1**	**0**	0
1	0	1	1	0

25. Introduzca 2 errores aleatorios de un bit en el mensaje del ejercicio anterior. Indica el procedimiento para detectar los errores, utilizando dicho ejemplo.

El código a enviar es 110110110010110.

Si se introducen dos errores aleatorios, el código puede ser: 110110011010110.

Para detectar errores se crea la matriz con el mismo tamaño que se ha creado y se comprueba si coinciden los bits de paridad:

1	1	0	1	1
0	0	1	1	0
1	0	1	**1**	0

Ha habido un error en la paridad VRC, por lo que no es válido.

26. ¿Qué es un código de verificación de cuenta fija? Expón un ejemplo de código verificación de cuenta fija que permita detectar 1 error si el código tiene M=2 palabras.

Es un código uniforme donde todas sus palabras tienen el mismo número de 1s. Se llama i en n bits al número de *1s* en n bits que tiene cada mensaje. El resto de bits se encuentran a 0.

Ejemplo:

Símbolo	Código
A	00000001
B	00000010

Cumple los requisitos para ser un código de verificación de cuenta fija y además la distancia del código es 2, por lo que permite detectar 1 error.

27. ¿Cómo se decodifica un código de verificación de cuenta fija? ¿Cómo se detectan errores? Expón un ejemplo con una de las palabras del código diseñado en el ejercicio anterior.

Primero se debe detectar si existe un error. Para eso se cuenta el número de *1s* del mensaje. Si el número de *1*s coincide con el establecido, puede que esté correcto.

Cuando se sabe que el número de *1s* es el correcto, se decodifica el mensaje utilizando la tabla de codificación.

Ejemplo: se recibe el mensaje 00000001. Se comprueba el número de *1s*. Como el número de *1s* es 1 y coincide con el establecido, en principio es correcto. Se consulta en la tabla de codificación y se obtiene el símbolo *A*.

28. ¿Es posible diseñar un código de verificación de cuenta fija que permita detectar 3 errores? Justifica tu respuesta y, si es posible, pon un ejemplo de un código que cumpla con esta condición.

Si el código de verificación tiene distancia 4, podría detectar 3 errores. Si se tiene un código de verificación de *4* en *8*, por ejemplo, con *2* palabras se tendría:

- 00001111.
- 11110000.

En este caso, su distancia es 4, por lo que podría detectar 3 errores. Si se quisieran codificar más palabras ya no sería válido.

29. Explica cuál es la correspondencia entre un polinomio de grado *n* con coeficientes en Z_2 valores {0,1}), y un código binario.

Existe una transformación directa, de manera que los coeficientes valen 0 o 1 dependiendo del valor del polinomio. En este caso, se puede decir que el polinomio $x^2 + x^0$ es de grado 2, tiene 3 coeficientes y, por tanto, el código binario es de 3 bits, siendo 101.

30. ¿Qué es un código CRC? ¿En qué se basan los códigos CRC?

Es un código que se representa como un polinomio. Al bit menos significativo se le asigna el menor coeficiente, al más significativo se le asigna el mayor coeficiente.

Modelan códigos con el esquema: $b3x^3 + b2x^2 + b1x^1 + b0x^0$.

Si se quiere un código de 4 bits, el grado del polinomio es 3. Si se quiere un código de 8 bits, el grado del polinomio es 7. Los coeficientes solo pueden ser valores 0 o 1.

Se basan en incluir redundancia. Cambia la forma en la que se calcula la redundancia y dónde se añade.

31. ¿Qué es un polinomio generador?

Es el polinomio que va a generar la redundancia. Es el código resultante de convertir el polinomio a conjunto de bits.

Es un polinomio que se utiliza para dividir el código y obtener un resto, que serán los bits de control de errores.

32. ¿Cómo se decodifica y detectan errores en el receptor, usando un código CRC?

Se concatena el código que se quiere enviar con el resto de dividir el código que se quiere enviar (con tantos *0s* al final como tamaño tenga el polinomio generador) y el polinomio generador.

Una vez que el receptor recibe el código, comprueba si el mensaje recibido es divisible por el polinomio generador. Si el resto es 0, no hay errores. Si es distinto de 0, hay errores en la transmisión.

Emisor y receptor disponen del polinomio generador.

33. Supongamos que se desea enviar el mensaje 1101 codificado en un código CRC con polinomio generador p(x) = $x^4 + 1$. ¿Cuántos bits tendrá el código CRC que se enviará? ¿Cómo se codifica el mensaje en el código CRC? Calcula el código CRC que se enviaría.

El código tendrá 4 bits (1101) más tantos *0s* como tenga el grado del polinomio generador. 11010000 será el mensaje a enviar. Tendrá 8 bits.

- Paso 1: se añaden tantos *0s* al final al mensaje que se desea enviar como el grado del polinomio generador: 1101**0000**.
- Paso 2: se divide por el polinomio generador (10001) y se le añade al mensaje inicial el resto de la división. Como el resto es 0100, el mensaje a enviar es **11010100**.

34. Introduce un error de 1 bit en el código calculado en el ejercicio anterior. Explica y calcula cómo el receptor detectaría el error en el mensaje.

El código del ejercicio anterior con un error de 1 bit es: 11010110.

Para comprobar si existe un error en el mensaje se debe dividir entre el polinomio generador. Si el resto es 0, no se detectaría ningún error. Si no es 0, existe algún error.

11010110 / 10001 = 1100. Resto: 1010. Como el resto no es 0, hay algún error.