

Decodificación de códigos LDPC: Algoritmo suma-producto

Procesamiento de señal en sistemas de comunicaciones y
audiovisuales

Máster Universitario en Ingeniería de Telecomunicación

ETSIT-UPV

Práctica 5

Índice

1. Objetivos del trabajo	3
2. Algoritmo suma-producto	3
3. Implementación	6

1. Objetivos del trabajo

Con este trabajo se pretende que los alumnos programen un decodificador LDPC implementando el algoritmo suma-producto estudiado en la teoría. Para verificar el correcto funcionamiento del algoritmo programado se hará uso de una pequeña matriz de comprobación de paridad proporcionada en la práctica.

2. Algoritmo suma-producto

El código LDPC se define en general como una matriz de comprobación de paridad, y puede ser expresado utilizando un grafo bipartito, que se denomina como grafo de Tanner. El grafo bipartito es un grafo en el que los vértices que constituyen el grafo están divididos en dos tipos diferentes, y el código LDPC está representado por el grafo bipartito que consiste en vértices, de los que unos se denominan nodos variables (v_i) y los otros nodos de comprobación (c_i).

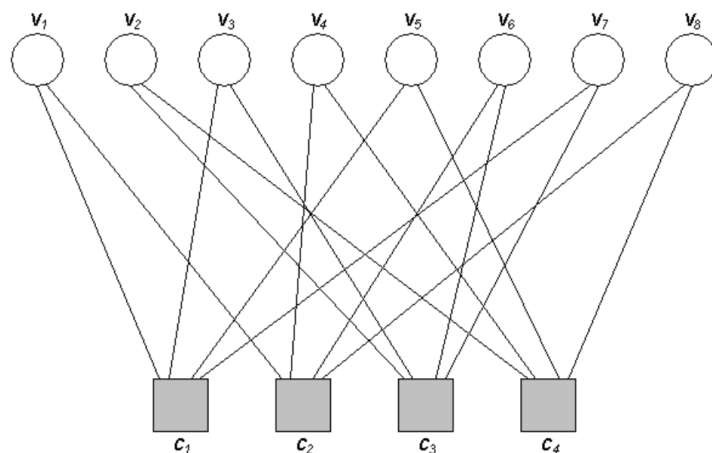


Figura 1: Grafo de Tanner.

Los nodos variables están mapeados uno a uno a los bits codificados. La decodificación eficiente de los códigos LDPC se logra mediante algoritmos iterativos que hacen un trabajo lineal en la cantidad de aristas y dado que la cantidad de aristas es $O(n)$, la complejidad de estos algoritmos resulta $O(n)$. En general, estos algoritmos se pueden ver como algoritmos de envío de mensajes, llamados así porque en cada iteración se envía un mensaje desde cada nodo de comprobación a cada variable relacionada y luego un mensaje de cada variable a cada nodo de comprobación relacionado. El algoritmo suma-producto es el algoritmo de decodificación de paso de mensajes de códigos LDPC más popular, ya que sus prestaciones se aproximan asintóticamente a las óptimas. Aunque existen varias simplificaciones del algoritmo, continua siendo considerado una referencia en la literatura. Es conveniente representar los mensajes que se pasan como estimaciones de log-likelihood ratios (LLRs). Consideramos para ello el caso donde la fuente de información binaria mapea los bits de la palabra código como $M(0) = -1$ y $M(1) = +1$. El funcionamiento del algoritmo suma-producto es el siguiente:

■ **Paso 1: Inicialización.**

En la fase de inicialización, el algoritmo determina los valores iniciales de las variables para cada uno de los nodos de variable y de los nodos de comprobación. La inicialización se realiza como sigue para cada uno de ellos:

$$v_i = \frac{-2r_i}{\sigma^2} \quad (1)$$

$$\lambda_{v_i \rightarrow c} = 0 \quad (2)$$

$$\lambda_{c_j \rightarrow v} = 0 \quad (3)$$

donde r_i representa el valor recibido para el simbolo i y $\sigma^2 = N_o/2$ es la varianza de ruido gaussiano del canal.

■ **Paso 2. Actualización de cada nodo de comprobación.**

En este estado, todos los nodos de chequeo serán actualizados mediante el paso de mensajes desde los nodos variables a través de las conexiones. Todos los nodos variables conectados a un cierto nodo de comprobación, mandan sus probabilidades. De este modo, el nodo de comprobación recoge esos mensajes y actualiza sus variables como sigue:

$$\lambda_{v_i \rightarrow c} = v_i + \sum_{c_j \in A(v_i) \setminus c} \lambda_{c_j \rightarrow v_i}. \quad (4)$$

donde $\lambda_{v_i \rightarrow c}$ representa el mensaje que va desde el nodo variable v_i hasta el nodo de comprobación c .

■ **Paso 3. Parada o continuación.**

En este estado, se decide si el algoritmo continua iterando, se satisface la condición de salida o se ha llegado al número máximo de iteraciones. El criterio de parada más común para el proceso del algoritmo suma-producto es la regla de palabra código valida. Este criterio de parada puede ser descrito como

$$\prod_{v_i \in A(c_j)} \lambda_{v_i \rightarrow c_j} > 0. \quad (5)$$

Si la expresión anterior es verdadera para todos los nodos de comprobación c_j , el algoritmo es finalizado. Esto es, si todos los nodos de comprobación de paridad son satisfechos tenemos una palabra código valida. Este parece ser un buen criterio de parada, sin embargo se pueden cometer errores puesto que el algoritmo puede converger a palabras código no válidas. Sin embargo, este escenario no es muy común y consecuentemente no es un gran problema. De este modo, si no hemos llegado al límite máximo de iteraciones y el criterio de parada no es satisfecho, continuamos al paso 4. Por el contrario, si hemos llegado al número máximo de iteraciones o el criterio de parada ha sido satisfecho continuamos al paso 5.

■ **Paso 4. Actualización de cada nodo variable.**

En este paso los nodos de chequeo toman los mensajes recibidos de cada nodo variable a los que están conectados y calculan la actualización de sus variables de la siguiente forma

$$\lambda_{c_j \rightarrow v} = 2 \tanh^{-1} \left(\prod_{v_j \in A(c_i) \setminus v} \tanh \left(\frac{\lambda_{v_i \rightarrow c_j}}{2} \right) \right). \quad (6)$$

Cuando esta computación es completada, se vuelve al paso 2.

■ **Paso 5. Salida.**

El algoritmo suma-producto ha finalizado su proceso iterativo y el último paso es calcular el correspondiente resultado de salida:

$$v'_i = v_i + \sum_{v_j \in A(c_i)} \lambda_{c_j \rightarrow v_i} \quad (7)$$

donde v'_i representa el valor de probabilidad posteriori para cada simbolo de la palabra código en el dominio log-likelihood. Este valor es obtenido mediante las contribuciones de todos los nodos de chequeo

así como el valor original de v_i .

Finalmente la secuencia de información decodificada es obtenida mediante la aplicación de una decisión hard en los valores anteriormente calculados que corresponden con símbolos de información. Sabiendo que los valores de información con valores 0,1 han sido modulados como -1,+1 antes de la transmisión, después del proceso de decodificación cada simbolo m'_i estimado toma el valor acorde a

$$v'_i > 0 \quad \Leftrightarrow m'_i = 0 \quad (8)$$

$$v'_i < 0 \quad \Leftrightarrow m'_i = 1 \quad (9)$$

3. Implementación

Para realizar la práctica usaremos una matrix de chequeo de paridad \mathbf{H} de pequeñas dimensiones. Esta matrix será utilizada para comprobar el correcto funcionamiento del algoritmo suma-producto programado.

La matriz a utilizar será la siguiente

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

la cual corresponde al siguiente grafo mostrado en la Figura 1.

Empezaremos construyendo la palabra código \mathbf{c} la cual queremos transmitir. Para todo código bloque lineal, la palabra código todo ceros es una palabra código valida, de ahí que vayamos a asumir que vamos a transmitir el siguiente mensaje

$$\mathbf{m} = [0 \ 0 \ 0 \ 0]$$

la cual corresponde con la palabra código

$$\mathbf{c} = \mathbf{mG} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0].$$

La estructura de la matriz de chequeo de paridad \mathbf{H} utilizada indica que el mensaje de información consiste en 4 bits y la palabra código contiene 8.

Con respecto a la suposición anterior, no necesitamos conocer la matrix generadora \mathbf{G} (la cual sería necesaria en caso de que los elementos de \mathbf{m} fuesen diferentes).

Por otro lado, vamos a asumir una modulación BPSK, de este modo los valores de una palabra código son mapeados en los valores +1 -1 antes

de ser transmitidos a través del canal. De este modo, tras el proceso de modulación y debido a al mapelo $[0, 1] \rightarrow [-1, +1]$, la palabra código todo ceros corresponde a

$$\mathbf{c}_{\text{mod}} = [-1 \quad -1 \quad -1 \quad -1 \quad -1 \quad -1 \quad -1 \quad -1] \quad (10)$$

Ahora, asumimos que \mathbf{c}_{mod} es enviado a través de un canal ruidoso que da lugar a la palabra recibida \mathbf{r} en el receptor:

$$\mathbf{r} = [-2,5467 \quad 0,2358 \quad -1,3929 \quad -3,0287 \quad -1,8290 \quad -1,1768 \quad -1,9434 \quad -0,1152] . \quad (11)$$

Dado \mathbf{r} vamos a seguir los pasos del algoritmo suma-producto para decodificar el mensaje enviado:

- El primer paso del algoritmo es inicializar los símbolos de la palabra código. Tenemos que inicializar todas las variables de la palabra código con su correspondientes valor a priori. La inicialización se realiza acorde a (1). Para ello, es asumido que la varianza del ruido del canal es conocida (a veces descrita como $\sigma^2 = N_0/2$ para un canal de ruido blanco aditivo gaussiano (AWGN)). En nuestro caso asumimos transmisión a través de un canal AWGN cuya varianza es $N_0/2 = 1$.
- Salida del Paso 2 (Primera iteración) (ver ecuacion (4))

Desde nodo Variable 1 al nodo Comprobación 1 = 5.0935
 Desde nodo Variable 1 al nodo Comprobación 2 = 5.0935
 Desde nodo Variable 2 al nodo Comprobación 3 = -0.4716
 Desde nodo Variable 2 al nodo Comprobación 4 = -0.4716
 Desde nodo Variable 3 al nodo Comprobación 1 = 2.7859
 Desde nodo Variable 3 al nodo Comprobación 3 = 2.7859
 Desde nodo Variable 4 al nodo Comprobación 2 = 6.0574
 Desde nodo Variable 4 al nodo Comprobación 4 = 6.0574
 Desde nodo Variable 5 al nodo Comprobación 1 = 3.6580
 Desde nodo Variable 5 al nodo Comprobación 4 = 3.6580
 Desde nodo Variable 6 al nodo Comprobación 2 = 2.3535
 Desde nodo Variable 6 al nodo Comprobación 3 = 2.3535
 Desde nodo Variable 7 al nodo Comprobación 1 = 3.8867
 Desde nodo Variable 7 al nodo Comprobación 3 = 3.8867
 Desde nodo Variable 8 al nodo Comprobación 2 = 0.2305
 Desde nodo Variable 8 al nodo Comprobación 4 = 0.2305

- Salida del Paso 3 (Primera iteración) (ver ecuacion (5))

Satisfacción del nodo Comprobación 1=201.7455

Satisfacción del nodo Comprobación 2=16.7372

Satisfacción del nodo Comprobación 3=-12.0177

Satisfacción del nodo Comprobación 4=-2.4086

Debido a que no todos los nodos de Comprobación cumplen la condición, no tenemos una palabra código válida y por ello se continua la iteración.

- Salida del Paso 4 (Primera iteración) (ver ecuación (6))

Desde el nodo Comprobación 1 al nodo Variable 1 = 2.2290

Desde el nodo Comprobación 2 al nodo Variable 1 = 0.1893

Desde el nodo Comprobación 3 al nodo Variable 2 = 1.7387

Desde el nodo Comprobación 4 al nodo Variable 2 = 0.2178

Desde el nodo Comprobación 1 al nodo Variable 3 = 2.9490

Desde el nodo Comprobación 3 al nodo Variable 3 = -0.3715

Desde el nodo Comprobación 2 al nodo Variable 4 = 0.1879

Desde el nodo Comprobación 4 al nodo Variable 4 = -0.0505

Desde el nodo Comprobación 1 al nodo Variable 5 = 2.4284

Desde el nodo Comprobación 4 al nodo Variable 5 = -0.0529

Desde el nodo Comprobación 2 al nodo Variable 6 = 0.2266

Desde el nodo Comprobación 3 al nodo Variable 6 = -0.3980

Desde el nodo Comprobación 1 al nodo Variable 7 = 2.3708

Desde el nodo Comprobación 3 al nodo Variable 7 = -0.3415

Desde el nodo Comprobación 2 al nodo Variable 8 = 2.2689

Desde el nodo Comprobación 4 al nodo Variable 8 = -0.4449

- Salida del Paso 2 (Segunda iteración) (ver ecuación (4))

Desde nodo Variable 1 al nodo Comprobación 1 = 5.2828

Desde nodo Variable 1 al nodo Comprobación 2 = 7.3224

Desde nodo Variable 2 al nodo Comprobación 3 = -0.2538

Desde nodo Variable 2 al nodo Comprobación 4 = 1.2671

Desde nodo Variable 3 al nodo Comprobación 1 = 2.4144

Desde nodo Variable 3 al nodo Comprobación 3 = 5.7348

Desde nodo Variable 4 al nodo Comprobación 2 = 6.0069

Desde nodo Variable 4 al nodo Comprobación 4 = 6.2453

Desde nodo Variable 5 al nodo Comprobación 1 = 3.60865
 Desde nodo Variable 5 al nodo Comprobación 4 = 6.0865
 Desde nodo Variable 6 al nodo Comprobación 2 = 1.9556
 Desde nodo Variable 6 al nodo Comprobación 3 = 2.5801
 Desde nodo Variable 7 al nodo Comprobación 1 = 3.5452
 Desde nodo Variable 7 al nodo Comprobación 3 = 6.2575
 Desde nodo Variable 8 al nodo Comprobación 2 = -0.2144
 Desde nodo Variable 8 al nodo Comprobación 4 = 2.4994

■ Salida del Paso 3 (Segunda iteración) (ver ecuación (5))

Satisfacción del nodo Comprobación 1=163.0174
 Satisfacción del nodo Comprobación 2=-18.4419
 Satisfacción del nodo Comprobación 3=-23.4991
 Satisfacción del nodo Comprobación 4=120.3822

Debido a que no todos los nodos de Comprobación cumplen la condición, no tenemos una palabra código válida y por ello se continua la iteración.

■ Salida del Paso 4 (Primera iteración) (ver ecuación (6))

Desde el nodo Comprobación 1 al nodo Variable 1 = 1.9331
 Desde el nodo Comprobación 1 al nodo Variable 1 = -0.1602
 Desde el nodo Comprobación 1 al nodo Variable 2 = 2.5148
 Desde el nodo Comprobación 1 al nodo Variable 2 = 2.4497
 Desde el nodo Comprobación 1 al nodo Variable 3 = 2.7959
 Desde el nodo Comprobación 1 al nodo Variable 3 = -0.2169
 Desde el nodo Comprobación 1 al nodo Variable 4 = -0.1608
 Desde el nodo Comprobación 1 al nodo Variable 4 = 1.0285
 Desde el nodo Comprobación 1 al nodo Variable 5 = 2.0957
 Desde el nodo Comprobación 1 al nodo Variable 5 = 1.0293
 Desde el nodo Comprobación 1 al nodo Variable 6 = -0.2131
 Desde el nodo Comprobación 1 al nodo Variable 6 = -0.2512
 Desde el nodo Comprobación 1 al nodo Variable 7 = 2.1092
 Desde el nodo Comprobación 1 al nodo Variable 7 = -2163
 Desde el nodo Comprobación 1 al nodo Variable 8 = 1.9342
 Desde el nodo Comprobación 1 al nodo Variable 8 = 1.2535

■ Salida del Paso 2 (Tercera iteración) (ver ecuación (4))

Desde nodo Variable 1 al nodo Comprobación 1 = 4.9333
 Desde nodo Variable 1 al nodo Comprobación 2 = 7.0265
 Desde nodo Variable 2 al nodo Comprobación 3 = 1.9781
 Desde nodo Variable 2 al nodo Comprobación 4 = 2.0432
 Desde nodo Variable 3 al nodo Comprobación 1 = 2.5690
 Desde nodo Variable 3 al nodo Comprobación 3 = 5.5817
 Desde nodo Variable 4 al nodo Comprobación 2 = 7.0859
 Desde nodo Variable 4 al nodo Comprobación 4 = 5.8966
 Desde nodo Variable 5 al nodo Comprobación 1 = 4.6874
 Desde nodo Variable 5 al nodo Comprobación 4 = 5.7537
 Desde nodo Variable 6 al nodo Comprobación 2 = 2.1023
 Desde nodo Variable 6 al nodo Comprobación 3 = 2.1405
 Desde nodo Variable 7 al nodo Comprobación 1 = 3.6704
 Desde nodo Variable 7 al nodo Comprobación 3 = 5.9960
 Desde nodo Variable 8 al nodo Comprobación 2 = 1.4840
 Desde nodo Variable 8 al nodo Comprobación 4 = 2.1647

- Salida del Paso 3 (Segunda iteración) (ver ecuación (5))

Satisfacción del nodo Comprobación 1=218.0386
 Satisfacción del nodo Comprobación 2=155.3315
 Satisfacción del nodo Comprobación 3=141.7045
 Satisfacción del nodo Comprobación 4=150.0543

Debido a que todos nodos de Comprobación cumplen la condición de parada, ¡hemos encontrado una palabra código valida! El siguiente paso es obtener la representación binaria de la palabra código y el mensaje.

- Salida del Paso 5

Calculamos los valores de salida acorde a ecuación (7)

$$\mathbf{v}' = [6,8664 \quad 4,4929 \quad 5,3648 \quad 6,9251 \quad 6,7831 \quad 1,8893 \quad 5,7796 \quad 3,4181]$$

y las decisiones hard con respecto a estos valores son dados por (8)

$$\mathbf{c} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Finalmente, dada la suposición que la primera parte de la palabra código es la que corresponde con el mensaje, obtenemos \mathbf{m}' como

$$\mathbf{m}' = [0 \quad 0 \quad 0 \quad 0]$$