

Métodos Numéricos Avanzados

Un sistema de comunicaciones

María de la Puerta Echeverría (50009), Teresa Fontanella De Santis (52455) y Tomás Mehdi (51014)
 Grupo 2 - Trabajo Práctico 1 - Instituto Tecnológico de Buenos Aires (ITBA)

Resumen—En el presente artículo se describe la estimación de un canal de un sistema de comunicación mediante el método de cuadrados mínimos. El entorno de programación que se utiliza es Matlab.

Palabras claves—Sistema de comunicación, estimación de canal, respuesta al impulso, cuadrados mínimos, QR.

1 INTRODUCCIÓN

UN sistema de comunicaciones consta, como mínimo, de tres componentes: un transmisor (que envía los datos), un receptor (que los recibe) y el medio por donde se transmiten los datos o *canal*. Dado que este último modifica la información transmitida, es necesario que el receptor pueda conocerlo, para poder interpretar los datos de forma adecuada. Como generalmente no se conoce, resulta necesario estimarlo. A eso se le llama *estimación del canal*. En el presente informe se estima el canal, utilizando el método QR para cuadrados mínimos.

En la siguiente sección, Metodología Utilizada, se describe la misma, así como las pruebas realizadas. En la última sección se muestran los resultados obtenidos y se extraen conclusiones.

2 METODOLOGÍA UTILIZADA

Para la realización de este trabajo, se toman en cuenta las siguientes etapas:

2.1 Modelo empleado

El transmisor envía un dato s_k cada T segundos, donde s_0 es enviado en $t = 0$, s_1 en $t = T$, s_2 en $t = 2T$, ...

La modificación de los datos por el canal se presenta por la denominada *respuesta al impulso del canal* $\{h_k\}_{k=0}^L$, donde L es la longitud de la respuesta al impulso.

Además de ser modificados por el canal, los datos son afectados por ruido blanco Gaussiano aditivo $N_k \sim N(0, \sigma)$.

Teniendo todo esto en cuenta, cada T segundos el receptor observa:

$$r_n = \sum_{k=0}^{L-1} h_k s_{n-k} + N_n. \quad (1)$$

También podemos expresar la Ec. (1) en forma matricial como

$$\vec{r} = H\vec{s} + \vec{N}, \quad (2)$$

donde

$$\vec{r} = \begin{pmatrix} r_0 \\ r_1 \\ \dots \\ r_M \end{pmatrix}, \vec{s} = \begin{pmatrix} s_0 \\ s_1 \\ \dots \\ s_{M-1} \end{pmatrix}, \vec{N} = \begin{pmatrix} N_0 \\ N_1 \\ \dots \\ N_{M-1} \end{pmatrix}, \quad (3)$$

$$H = \begin{pmatrix} h_0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ h_1 & h_0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & 0 & 0 & \dots & 0 & 0 \\ h_{L-1} & h_{L-2} & \dots & h_1 & h_0 & 0 & \dots & 0 & 0 \\ 0 & h_{L-1} & \dots & h_2 & h_1 & h_0 & \dots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & h_1 & h_0 \end{pmatrix} \in \mathbb{R}^{M \times M}, \quad (4)$$

y $M \geq L$.

Otra forma equivalente es la siguiente:

$$\vec{r} = S\vec{h} + \vec{N}, \quad (5)$$

donde \vec{r} y \vec{N} son como antes y

$$\vec{h} = \begin{pmatrix} h_0 \\ h_1 \\ \dots \\ h_L \end{pmatrix}, \quad (6)$$

$$S = \begin{pmatrix} s_0 & 0 & \dots & \dots & 0 \\ s_1 & s_0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ s_{L-2} & s_{L-3} & \dots & s_0 & 0 \\ s_{L-1} & s_{L-2} & \dots & s_1 & s_0 \\ s_L & s_{L-1} & \dots & s_2 & s_1 \\ \vdots & s_{M-1} & s_{M-2} & \dots & s_{M-L} \end{pmatrix} \in \mathbb{R}^{M \times L}. \quad (7)$$

Lo que nos interesa es recuperar correctamente la señal enviada \vec{s} a partir de la señal recibida \vec{r} . Esto no sería tan difícil si se conociesen L y los $\{h_k\}$. En efecto, tomando $M = L$ en la Ec. (2), podemos buscar \vec{s} que minimice:

$$\|H\vec{s} - \vec{r}\|_2^2. \quad (8)$$

El problema es, sin embargo, que en general ni L ni $\{h_k\}$ son conocidos: estimarlos corresponde al problema de estimación del canal. Si L es conocido, una forma de estimar el canal es mediante la Ec. (5) y el método de cuadrados mínimos. En efecto, tómesese $M > L$ y envíese una señal conocida por el receptor $\vec{s} \in \mathbb{R}^M$ (denominada secuencia de entrenamiento). Luego, se estima el canal planteando el siguiente problema de cuadrados mínimos: encontrar $\vec{h} \in \mathbb{R}^L$ que minimice

$$\|S\vec{h} - \vec{r}\|_2^2. \quad (9)$$

2.2 Pruebas realizadas

Para efectuar las pruebas correspondientes, se considera utilizar un $L = 1, 10, 30, 50$ ¹, aplicando un ruido gaussiano $\sigma = 1$. Se considera este valor por ser lo suficientemente pequeño para no distorsionar demasiado los datos a transmitir. También se prueba con diferentes longitudes de la cadena de entrenamiento $M = 32, 512, 1024$.

Para analizar cómo se estima en "condiciones óptimas" (verbigracia: sin ruido, etc.), también se prueba con $M = 512$ y $\sigma = 0$ (con los valores de L ya mencionados)².

1. Resolviendo dudas con el profesor, se ha llegado a la conclusión de que a valores de L mayores a 10, el resultado obtenido no es satisfactorio. Por este motivo, para que la prueba sea exitosa los valores de L deben ser $L = 1, 2, 3, 4, 5$.

2. Debido a que, en la implementación, la imagen se transmitía con $L = 30$ (valor bastante distinto a los valores de L usados para recuperar la información), los resultados obtenidos eran dispares e inconexos. Se acordó con el profesor que, en vez de rehacer todo el trabajo, agregar las pruebas realizadas con este nuevo cambio, en el cual la información se transmite con $L = 5$, usando $\sigma = 1$, $M = 512$ y se considera a la hora de recibir los datos $L = 1, 2, 3, 4, 5$.



Fig. 1: Imagen transmitida (arriba) y recibida (abajo) usando $L = 1, M = 512, \sigma = 1$.



Fig. 2: Imagen transmitida (arriba) y recibida (abajo) usando $L = 3, M = 512, \sigma = 1$.

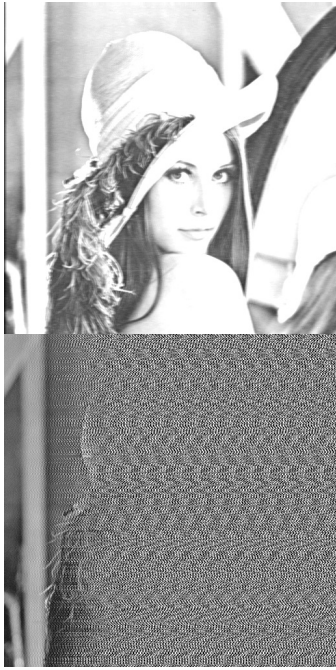


Fig. 3: Imagen transmitida (arriba) y recibida (abajo) usando $L = 5, M = 512, \sigma = 1$.

3 RESULTADOS Y CONCLUSIONES

Para poder cuantificar la confiabilidad de una imagen recuperada, se ha pensado en un índice i que consiste en

$$i = \frac{\text{cantidad_bits_iguales}}{\text{cantidad_bits_imagen}}.$$

TABLA 1:
Relación entre imagen transmitida e imagen recibida

L M	32	64	256	512
1	0.0019798	0.082809	0	0.0048752
2	0.22441	0	0.0010986	0.029369
3	0	0.0024719	0.12458	0.0026894
4	0.00020218	0.000045776	0.0062218	0.000030518
5	0.088345	0.00068665	0.00061417	0.18251

TABLA 2:
Relación entre imagen recibida e imagen original

L M	32	64	256	512
1	0.0020866	0.0015945	0.0031052	0.0053940
2	0.00079727	0.029865	0.023960	0.000061035
3	0	0.0011215	0.00055695	0.00011826
4	0.000030518	0.00029755	0.059250	0.000057220
5	0.00022125	0.10743	0.00088501	0.0099983

Analizando tanto la Tabla 1, como la Tabla 2 puede decirse que completamente distintas. Antes de la corrección efectuada, era imposible afirmar que este método dependiera de que el ruido sea menor para obtener una recuperación exitosa, ya que se observaban situaciones en las que se obtiene una imagen muy distorsionada, habiendo transmitido una imagen con

poco ruido. Después, con el último set de pruebas realizadas, se puede afirmar que tomando valores muy altos de L , la imagen no se recupera bien, pero que, para el resto de los valores de L , se recupera de manera aceptable. Esto puede ocurrir debido a que, aún cuando el L estimado sea igual al real, los errores de precisión provocan imágenes distorsionadas (como ocurre con $L = 5$). Sin embargo, se puede concluir que este método puede usarse para la estimación de un canal de comunicaciones, para L chicos.

BIBLIOGRAFÍA

- [1] John Matthews and Kurtis Fink, *Métodos Numéricos con Matlab*, Fecha de publicación: Mayo 2000 — ISBN-10: 8483221810 — ISBN-13: 978-8483221815 — Edición: 3
- [2] Apuntes de la Cátedra.
- [3] Consultas a la Cátedra, a quien agradecemos su gentil colaboración, por su experta ayuda en la preparación de este artículo.

APÉNDICE

.1 Código y ejecución

```
function ej4(E, L, sigma)

%E tamaño del s de entrenamiento (es un vector)
%E > L
%Analizar casos en que E es distinto que cols(imagen) 32,1024

%PASO 1
'Enviando imagen por canal inestable'
%h random para el primer caso en el que envío s de entrenamiento
L_aux = 5;
ganancia = 1/5;
h = ganancia*(1+randn(L_aux,1));

a = imread('lena512.bmp');
%a = 255*rand(10,10);
imwrite(uint8(a), 'realimg.bmp');
%a=eye(5,5);

%Agrego una fila conocida a la imagen de longitud M, cols(imagen)
sTrainSent = 255*[rand(1,E) zeros(1,size(a,2)-E)];
sTrainSent = double(sTrainSent);
a = [ a ; sTrainSent ];

%imshow(a); % muestro la imagen original

M = size(a,2);
P = size(a,1);
H = toeplitz([h.' zeros(1,M-L_aux)], zeros(1,M));
r = zeros(M,P);
N = sigma*randn(M,1); % ruido
%Envío la imagen con un s extra conocido para entrenando
%Se transmite una fila de la imagen por vez
for k=1:P
    s = double(a(k,:).'); % lo que se envía
    r(:,k) = H*s+N; % lo que se recibe
    N = sigma*randn(M,1); % mas ruido
end
b = uint8(r(:,1:P-1).');
%imshow(b);
imwrite(b, 'imgTrans.bmp');

% r = H*s+N
% r = S*h+N

%PASO 2
'Estimando H'
%Estimo h' x donde x es la longitud(L) de respuesta al impulso
%Tengo r y S, uso QR para estimar h
%||S*h-r||_{2}^{2}
S = toeplitz(sTrainSent, zeros(1,L)); % S
%Uso M para cortar sTrainSent por que si mando uno mas largo que
%cols(imagen) ya no me interesan los que sobran
sTrainReceived = double(r(:,P)); % r, lo que recibí del entrenamiento
```

```

[Q R] = ourQR(S); %  $S*h = r$  Entonces  $Q'*S*h = Q'*r$  Entonces  $R*h = Q'*r$ 
%[Q R] = ourQR(S); % Nuestra implementacion
h_estimada = ecuacionTriangularSolver(R,(Q'*sTrainReceived)); % Resolvemos  $R*h = Q'*r$  y usamos
H_estimada = toeplitz ([h_estimada.' zeros(1,M-L)],zeros(1,M)); % Obtenemos H con h

%PASO 3
'Recuperando'
%Recupero la imagen con el H_estimada
r2 = r.';
for k=1:P-1
    u(:,k) = H_estimada\r(:,k); %lo que se recupera
    u(:,k) = H_estimada\r(:,k); %lo que se recupera
end
%u(:,P+1) = sTrainReceived; %lo que recibí última línea
b2 = uint8(u. ');
%imshow(b2);
imwrite(b2, 'imgRec.bmp');

```

.2 Implementación QR completo por G-S

```

function [Q R] = ourQR (A)
[rows cols]=size(A);
Q(:,1)=A(:,1);
Q(:,1)=Q(:,1)/norm(Q(:,1),2);
R(:,1)=Q(:,1)'*A(:,1);
for i=2:rows
    if i > cols
        Q(:,i)=rand(3,1);
    else
        Q(:,i)=A(:,i);
    end
    for k=1:i-1
        Q(:,i) = Q(:,i) - ((Q(:,i))'*Q(:,k)) * Q(:,k);
    end
    Q(:,i) = Q(:,i)/norm(Q(:,i),2);
    R(:,1)=Q(:,i)'*A(:,i);
end
[Rows Rcols] = size(R);
R = [ R ; zeros(rows-Rows,Rcols) ];

end

```

.3 Función para resolver ecuaciones lineales

```

function z = ecuacionTriangularSolver (x, y)
answerQty=size(x)(2);

z=zeros(answerQty,1);
z(answerQty)=y(answerQty)/x(answerQty,answerQty);
for i=answerQty-1:-1:1
    aux=x(i,i+1:answerQty)*z(i+1:answerQty);
    z(i)=(y(i)-aux)/x(i,i);
end

endfunction

```

.4 Función para índice de confiabilidad

```
function z=confiability(recovered,original)
    z=0;
    recoveredCols=size(recovered)(2);
    recoveredRows=size(recovered)(1);
    for i=1:recoveredRows
        for j=1:recoveredCols
            if(recovered(i,j)==original(i,j))
                z=z+1;
            end
        endfor
    endfor
    z=z/(recoveredCols*recoveredRows);
end
```

.5 Otros datos representativos de las pruebas

h	h_estimada
0.0592715994972346	1.22138294001092
0.316475471117918	
0.323844578892922	
0.531842458909969	
0.402201195880913	

Tabla 1 : Valores correspondientes a la prueba con $L = 1, M = 512$, 1 de Fig.1

h	h_estimada
0.445541676764217	0.457982382433996
0.101034422381018	0.147135682171331
0.0637079819621977	0.0806792295209017
-0.132967712825480	
0.219679633044226	

Tabla 2 : Valores correspondientes a la prueba con $L = 3, M = 512$, 1 de Fig.2

h	h_estimada
0.0455337140835005	0.0449846995202603
0.108336297046597	0.108106879455770
0.149204486339813	0.149598377126117
0.453840022112420	0.453617070901002
0.607263144894805	0.608308008105862

Tabla 3 : Valores correspondientes a la prueba con $L = 5, M = 512$, 1 de Fig.3