

## Práctica 2.

### Modelado de la señal de voz como un proceso aleatorio, y su aplicación en la transmisión sobre un canal con pérdidas.

*Teresa de Jesús Algarra Ulierte*

#### Descripción:

Consideramos una señal de voz digitalizada muestreada a  $16KHz$  y 8 bits. El canal de transmisión usado pierde una muestra de señal cada 10 muestras. Vamos a modelar la señal de dos formas diferentes:

- Como un proceso i.i.d.(variables independientes e idénticamente distribuidas).
- Como un proceso de Markov, es decir, como una cadena de Markov.

El uso de modelos nos ayudará a reconstruir la señal que no ha llegado al receptor, para compensar las pérdidas del canal. Lo haremos de las siguientes maneras:

- Consideramos la señal como un proceso i.i.d. formado por variables discretas aleatorias que pueden tomar valores  $S(n) = m_1, m_2, \dots, m_{256} = -127, -126, \dots, 127, 128$ , es decir, un total de 256 valores incluido el 0. Hay que caracterizar el proceso, es decir, hallar las probabilidades  $P[S(n) = m_i]$  para  $i = 1, 2, \dots, 256$  usando los ficheros dados en el material de la práctica. Usaremos “aage3201R.8bi”, “tsge3270R.8bi”, “euge0013R.8bi” y “vege1850R.8bi”.
- Seguidamente vamos a considerar el modelo de Markov, que es más sofisticado. Aquí tenemos que determinar la matriz de probabilidades de transición formada por el conjunto de probabilidades condicionales  $P[S(n) = m_j | S(n-1) = m_i]$  para  $i = 1, 2, \dots, 256$ . Sale una matriz de probabilidades de orden 256. Usaremos los mismos ficheros que en el primer modelo.

#### Estimación:

Para suplir las muestras perdidas realizaremos cuatro tipos de estimaciones estadísticas.

- La primera estimación consistirá en sustituir  $s(n)$  por  $s'(n) = m_i$  que haga máxima  $P[s(n) = m_i]$ .

- La segunda estimación consistirá en sustituir  $s(n)$  por

$$s'(n) = \sum_{j=1}^{256} m_j P[s(n) = m_j]$$

- La tercera estimación esta está basada en el modelo de Markov. Sabiendo que en el instante  $n - 1$   $s(n - 1) = m_r$ , consistirá en sustituir  $s(n)$  por  $s'(n) = m_j$  que haga máxima  $P[s(n) = m_j | s(n - 1) = m_r]$ . Sabiendo que en el instante  $n - 1$   $s(n - 1) = m_r$ .
- La cuarta estimación esta está basada en el modelo de Markov. Sabiendo que en el instante  $n - 1$   $s(n - 1) = m_r$ , consistirá en sustituir  $s(n)$  por

## Evaluación:

Para ver cómo funciona cada una de las estimaciones propuestas, de cada vector de 10 muestras eliminaremos una aleatoriamente y la sustituiremos por la estimación. Lo haremos con el fichero “euge0019R.8bi”.

Para proceder a la evaluación objetiva de cada método, usaremos el indicador SNR tal que:

$$SNR_k(dB) = 10 \log_{10} \left[ \frac{\sum_{n=1}^N s^2(n)}{\sum_{n=1}^N (s(n) - s_k(n))^2} \right]$$

Donde N es el número total de muestras del fichero.

## Realización:

Empezamos la realización con el primer modelo. Tenemos que inicializar el programa eliminando los posibles restos de un programa anterior para que no de problemas.

```
clear all;
```

Después, abrimos los cuatro archivos que vamos a utilizar:

```
f1=fopen('vege1850R.8bi','r');
[s1, num1]=fread(f1, 'int8');
fclose(f1);
```

```
f2=fopen('aage3201R.8bi','r');
[s2, num2]=fread(f2, 'int8');
fclose(f2);
```

```

f3=fopen('tsge3270R.8bi','r');
[s3, num3]=fread(f3, 'int8');
fclose(f3);

f4=fopen('euge0013R.8bi','r');
[s4, num4]=fread(f4, 'int8');
fclose(f4);

```

Para escucharlos hacemos:

```
soundsc(s1,16000)
```

Tenemos que poner como argumento de la función 16000 porque es a la frecuencia de muestreo a la que está en KHz.

Seguidamente procedemos a concatenar todos los archivos binarios de audio en un vector para poder trabajar con la base de datos de manera cómoda:

```
S=[s1;s2;s3;s4];
```

Para su posterior uso, hallamos el numero de muestras de la base de datos con la función length:

```
total_muestras=length(S);
```

Para ver cuántas veces sale cada valor, podríamos hacer un bucle, pero es muy pesado, porque hay que repetir el mismo bucle 256 veces, ya que hay 256 valores distintos. Usamos una función ya definida de Matlab: los histogramas. Nos dice, dados intervalos, cuántas veces se dan números dentro de ese intervalo y nos da o bien una gráfica o bien en un vector.

%Definimos un vector x para que estén claros los intervalos de cada %muestra. Lo que haces es hacer intervalos [-127, -126.5], [-126.5, -125.5] %etc. Además, lo ploteamos para ver cómo va:

```
x=-127:128;
```

```
H=hist(S,x);
```

```
figure(1); plot(x,H);
```

%Pasamos a calcular la probabilidad, y ploteamos el resultado. Tiene que %ser igual que el histograma pero a escala más pequeña:

```
P=H / total_muestras;
```

```
figure(2); plot(x,P)
```

%Comprobamos que todas las probabilidades suman 1:

```
sum(P);
```

%Da 1, comprobado.

```

%OJO: ahora P(1)=P(S(n)=-127), ... , P(256)=P(S(n)=128). %Hay muchos
valores que vales 0, es decir, que no aparecen. Suelen ser las %de los extremos.
El problema es que al hacer ahora la ley de probabilidad, %nos va a decir que la
probabilidad de que S(n)=-127 es 0, cuando eso no es %posible, porque puede
aparecer, aunque ahora mismo en nuestra base de %datos no se haya dado por
ser muy limitada. Hay que corregirlo para que se %vea que los valores esos son
poco probables pero no imposibles. Para eso %fijamos un valor muy pequeño
sustituyendo a la fuerza las probabilidades 0 %por ese valor, en nuestro caso 10^(-
6). Así no nos sumará 1 la ley de %probabilidad, o sea que cuando lo hagamos
tenemos que renormalizar, es %decir, volver a hacer P=P/total_muestras y
sum(P):

for a=1:256; if P(a)==0; P(a)=10.^(-6); end end

sum(P);

%Lo planteamos para que se vea como los extremos de la gráfica son %diferentes:
figure(3); plot(x,P)

%Hay canales en los que se pierden, de cada 10 muestras, una. Tenemos
que %poner una estimación para que esa información no se pierda. Se puede
%sustituir por la muestra más probable. También se puede sustituir la %muestra
por el promedio de las muestras. Se llama MMS.

%Tenemos que simular el canal. OJO: cambio respecto del gión de prácticas.
%Usamos la función randi para eliminar una muestra de cada 10.

f5=fopen('euge0019R.8bi','r'); [s5, num5]=fread(f5, 'int8'); fclose(f5);

total_muestras2=length(s5);

H2=hist(s5,x);

figure(4); plot(x,H2);

P2=H2 / total_muestras2;

figure(5); plot(x,P2)

for a=1:256; if P2(a)==0; P2(a)=10.^(-6); end end

figure(6); plot(x,P2)

%Probamos a sustituir por la media de todo:

media=0;

for c=1:256; media=media+((-127+c)*P2(c)); end

media

for i=1:10:246; perdida=randi(10); P2(i-1+perdida)=media; end

figure(7); plot(x,P2)

```

```

SNR=0; media=round(media);
for c=1:256; SNR=SNR+((P2(c)-P2(media)).^2); end
SNR*8.69/2
%Pasamos a la estimacion 2: sustituimos por el valor mas probable:
[max, pos]=max(P2); sustitucion=pos;
for i=1:10:246; perdida=randi(10); P2(i-1+perdida)=P2(sustitucion); end
figure(8); plot(x,P2)

```