TEORÍA DE LA INFORMACIÓN

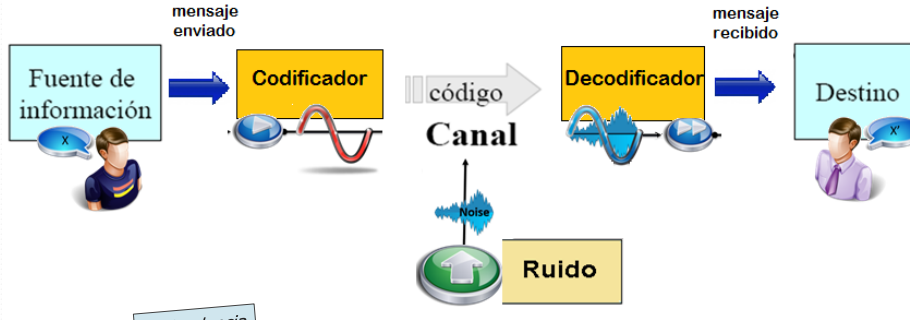
Tratamiento probabilístico de la Información

Fuentes de información

INGENIERÍA DE SISTEMAS

Cursada 2016

TRANSMISIÓN DE INFORMACIÓN



entropía

redundancia en los datos

técnicas de compresión

ruido en la transmisión

retransmisión de símbolos

pérdida de información

decodificación automática

errores en los datos recibidos

PRINCIPALES TEMÁTICAS
DE LA TEORÍA DE LA INFORMACIÓN

- ✓ Shannon propuso la Teoría de la Información y de la Comunicación
- ✓ sentó los fundamentos de la revolución tecnológica de la 2° mitad del siglo XX
- ✓ sus ideas posibilitaron la creación y proliferación de diversas tecnologías
  → diseño de circuitos digitales y computadoras, transmisión de datos, internet, métodos de compresión, criptografía, resolución de problemas en computadora, …

Claude Shannon (1916-2001)

¿Cómo medir la información? ¿Cuál es el contenido de información de un mensaje?

¿Cómo reducir la cantidad de datos necesarios para representar la información? → Compresión

¿Cuánta información puede transmitir un canal? → capacidad
¿Cómo transmitir información con mínima pérdida de información?

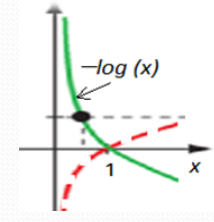
¿Cómo advertir si hay error en la recepción de un mensaje? ¿Es posible corregir los errores?

MEDIDA DE INFORMACIÓN

- Interesa determinar la probabilidad de aparición de cada mensaje –elegido entre todos los posibles– y su contenido de información (el aspecto semántico es irrelevante al problema ingenieril)
- Evento *x* con probabilidad de ocurrencia *P(x)*

  → contenido de información: (propuesto por Shannon)

$$I(x) = -\log P(x) = \log \frac{1}{P(x)}$$



- La *Teoría de la Información* se desarrolla en términos de probabilidades (naturaleza aleatoria o estocástica)

Propiedades :

- -log $P(x) \geq 0$, $0 \leq P(x) \leq 1$ (se considera log 0= 0) *La información nunca es negativa*
- $\lim_{P(x) \to 1}$ –log $P(x)$= 0 *Un evento seguro no aporta información*
- -log $P(x)$ > -log $P(y)$ si $P(x) < P(y)$ *Los eventos más probables aportan menor información (y viceversa)*
- -log $P(xy)$ = -log $(P(x) . P(y/x))$ = -log $P(x)$ – log $P(y/x)$

$\log_2$ (representación binaria)
→ Unidad: *BIT* (*BInary DigiT*)

MEDIDA DE INFORMACIÓN

Ejemplo

x= paridad del dado → P(x)= 1/2

$I(x)= -\log_2 (1/2) = 1$ bit

y= número del dado → P(y)= 1/6

$I(y)= -\log2 (1/6) \approx 2{,}58$ bit

¿qué significa I?

→ se asocia a la cantidad de preguntas binarias que deberían realizarse para conocer la salida del evento

→ se relaciona con la cantidad mínima de bits que se necesitan para representar esa información

La información obtenida luego de producirse un evento es equivalente a la incertidumbre que se tenía antes de conocer su resultado

FUENTES DE INFORMACIÓN

- Una fuente de información ***S*** genera en cada instante de tiempo *t* un símbolo s_i elegido dentro del conjunto de símbolos posibles, según su probabilidades de emisión (modelo)

Fuente de información → $s_1\ s_1\ s_2\ s_1\ s_3\ s_2 \ldots$

Estos símbolos pueden ser caracteres de un texto, pixels de una imagen, valores de una señal, etc.

- Clasificación:

➢ *Según el rango de valores que pueden generar:*

F. continua: puede generar símbolos dentro de un rango continuo de valores

F. discreta: puede generar símbolos dentro de un conjunto finito de valores

➢ *Según la relación entre sus símbolos:*

F. sin memoria (o de memoria nula): los símbolos son estadísticamente independientes

F. con memoria (de orden *k*): los símbolos son estadísticamente dependientes (de los *k* símbolos anteriormente emitidos)

FUENTES DE INFORMACIÓN

Ejemplo: aproximaciones al lenguaje Inglés
(Ej. de Shannon C.,
A Mathematical Theory of Communication)

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZL-HJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order approximation (digram structure as in English).

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TU-COOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONS-TURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

5. First-order word approximation. Rather than continue with tetragram, . . . , *n*-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NAT-URAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation. The word transition probabilities are correct but no further struc-ture is included.

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHAR-ACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

FUENTES SIN MEMORIA

- Si el modelo de probabilidades considerado describe una *fuente sin memoria* los símbolos son estadísticamente independientes → la probabilidad de emitir un símbolo no depende de los emitidos anteriormente
- La fuente se puede describir completamente mediante el conjunto de símbolos y sus probabilidades de ocurrencia
- Cada símbolo s_i posee una probabilidad de ocurrencia $p(s_i)$ y su contenido de información es $I(s_i)= -\log p(s_i)$

→ la cantidad media de información –*ENTROPÍA*– es:

$$\boxed{H(A) = -\sum_i P(A_i)\log P(A_i) = \sum_i P(A_i)\log\frac{1}{P(A_i)}}$$

cantidad promedio de preguntas binarias para conocer el resultado de *A*

ENTROPÍA DE FUENTE SIN MEMORIA

$$H(A) = -\sum_i P(A_i)\log P(A_i) = \sum_i P(A_i)\log\frac{1}{P(A_i)}$$

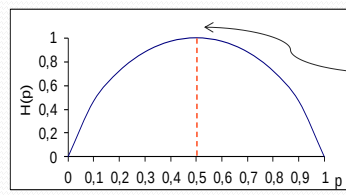
La entropía es un número real que depende de la distribución de probabilidades

Propiedades :

- *H(A)* ≥ 0 (La entropía nunca es negativa)
- *H(A)* = 0 sólo si uno de los eventos tienen prob. 1 y todos los demás prob. 0
- *H(A)* es una función continua y simétrica (respecto de las probabilidades)
- *H(A)* es máxima e igual a *log n* si los *n* eventos son equiprobables ($p_i = 1/n$, ∀i)

Caso de dos eventos de probabilidades:

{p, 1-p}



H(A) = log n
(entropía máxima)

Ejemplo

Una fuente X emite los símbolos 0, 1, 2
con probabilidades: 0.3, 0.5, 0.2, respectivamente

X= {0, 1, 2} p(x) = {0.3, 0.5, 0.2}

Contenido de información de los símbolos:

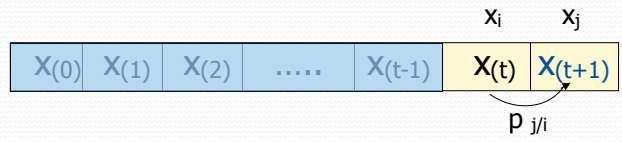
X	0	1	2
p(x)	0,3	0,5	0,2
I(x)	1,737	1	2,322

Entropía de la fuente *sin memoria*:

H(X)= - Σ p(x) log p(x) = -0.3 log 0.3 - 0.5 log 0.5 - 0.2 log 0.2 ≈ 1.485 bits

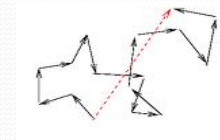
FUENTES CON MEMORIA

- Si se considera que los símbolos emitidos sucesivamente por la fuente de información son estadísticamente dependientes → la probabilidad de ocurrencia de un símbolo está condicionada por los anteriores
- Hipótesis de Markov → la probabilidad de emitir un símbolo depende sólo del símbolo emitido en el instante anterior



cadena de Markov

- Una fuente con memoria se puede modelar como un proceso estocástico markoviano de *random walk*



PROCESOS ESTOCÁSTICOS

- Un proceso estocástico es un fenómeno que evoluciona en el tiempo de manera impredecible, desde el punto de vista del observador
- En cada instante *t* se define una variable aleatoria que puede tomar diferentes valores (consideramos procesos discretos con un conjunto finito de estados)

Algunos ejemplos:

Señales (sonido, imágenes, video, ...)

clima (viento, humedad, temperatura,...)

Procesos económicos (acciones, stocks, PBI,...)

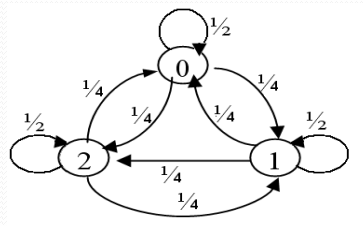
evolución de poblaciones, superficies, ...

FUENTES CON MEMORIA (ORDEN 1)

- En una fuente markoviana (con memoria de orden 1) la probabilidad de emitir un símbolo en *t+1* depende sólo del símbolo emitido en *t*
- Si las probabilidades de los símbolos se mantienen constantes en el tiempo → el proceso es markoviano *homogéneo*

Ejemplo

una fuente markoviana emite 3 símbolos 0, 1, 2 con las siguientes probabilidades de transición:



Grafo de transición de estados:
- Nodos: estados (símbolos)
- Arcos: transiciones posibles

FUENTES CON MEMORIA (ORDEN 1)

Las probabilidades condicionales de transición de estados se pueden disponer en una matriz de pasaje o de transición de estados:

$$M_{j/i} = \begin{array}{c} \\ s_1 \\ s_2 \\ \vdots \\ s_j \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} s_1 & s_2 & & s_i & \ldots \end{array} \\ \begin{bmatrix} p_{1/1} & p_{1/2} & \cdots & p_{1/i} & \cdots \\ p_{2/1} & p_{2/2} & \cdots & p_{2/i} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ p_{j/1} & p_{j/2} & \cdots & p_{j/i} & \ddots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{array}$$

probabilidades condicionales

$$\sum_j P(S(t+1)=s_j / S(t)=s_i) = \sum_j p_{j/i} = 1$$

VECTORES DE ESTADO

Para una fuente markoviana homogénea

- conocida la distribución de probabilidades de emisión en *t*, se puede obtener la distribución de probabilidades en *t+1* :

$$\mathbf{V}(t+1) = \mathbf{M}j/i \;.\; \mathbf{V}(t)$$

- dada la distribución inicial de probabilidades -**V**(0)- se pueden calcular los vectores de estado en cada uno de los instantes siguientes:

V(0) → condiciones iniciales del problema
V(1) = **M** .**V**(0)
V(2) = **M**. **V**(1) = **M. (M**.**V**(0)) = **M²**. **V**(0)
....
V(t+1) = **M**. **V**(t) = **M**$^{t+1}$.**V**(0)

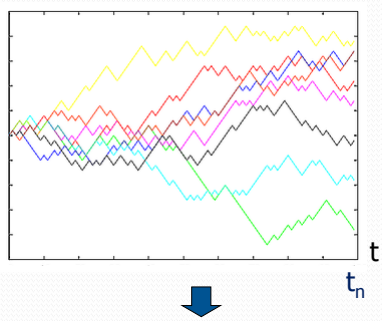
Ejemplo

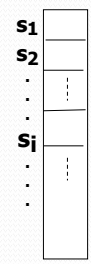
$$M = \begin{array}{c|ccc|} & 0 & 1 & 2 \\ 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 1 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 2 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{array}$$

	V0	V1	V2	V3	
$p_0(t)$	1	1/2	3/8	11/32	...
$P_1(t)$	0	1/4	5/16	21/64	...
$p_2(t)$	0	1/4	5/16	21/64	...

VECTORES DE ESTADO
por muestreo computacional

Generación de *trayectorias* de emisión de símbolos:





cada trayectoria representa un posible mensaje emitido por la fuente
(una realización del proceso estocástico)

obtener las componentes de **V(n)**:

$$\mathbf{P(V(n)=s_i)} \approx \frac{\#\ s_i \text{ en } n}{\#\text{total trayectorias}}$$

incorporar trayectorias a la simulación hasta que no se produzcan cambios significativos en los valores de probabilidad del vector

→ Verificar **convergencia**

VECTORES DE ESTADO
por muestreo computacional

Para el ejemplo anterior:

$$M = \begin{array}{c|ccc|} & 0 & 1 & 2 \\ \hline 0 & 1/2 & 1/4 & 1/4 \\ 1 & 1/4 & 1/2 & 1/4 \\ 2 & 1/4 & 1/4 & 1/2 \\ \hline & 0 & 1 & 2 \end{array}$$

$$Macum = \begin{array}{c} 0 \\ 1 \\ 2 \end{array} \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 3/4 & 3/4 & 1/2 \\ 1 & 1 & 1 \end{pmatrix}$$

condiciones iniciales (*según el problema*):

$$V_0 = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} \quad V0_{acum} = \begin{pmatrix} 1/3 \\ 2/3 \\ 1 \end{pmatrix}$$

```
PrimerSimb ()
 { r=rand()
   for(i=0 to 2)
     if (r<V0acum[i])
       return i; }
```

```
Sig_dado_Ant (col)
{r=rand()
  for(i=0 to 2)
    if ( r < Macum[i, col] )
      return i }
```

```
converge (A[ ], B[] )
{ for (i=0 to 2)
  { if (abs(A[i]-B[i]) > ξ )
      return FALSE }
  return TRUE }
```

```
Calcular_Vector_Estado (int n)
{
  emisiones=[0,0,0] // cant. de cada si
  Vn=[0,0,0]   //vector estado actual
  Vn_ant=[-1,0,0] //vector estado ant.
  #tray=0  //cant. trayectorias
  pasos  //cant. transiciones o pasos
 while not converge (Vn, Vn_ant) (1)
 {  s=PrimerSimb ();
    #tray++
    for (pasos= 0 to n)
       s=Sig_dado_Ant (s)
    emisiones[s]++
    Vn_ant ← Vn
    Vn ← emisiones/#tray
 } return Vn
}
```

(1) Se pueden agregar condiciones adicionales sobre cantidad de iteraciones

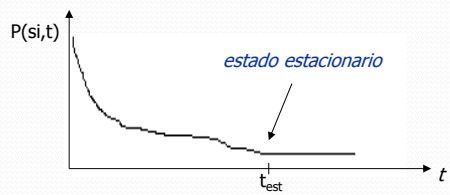
ESTADO ESTACIONARIO

- La distribución de probabilidades en cada *t* (vectores de estado) va variando con la evolución del proceso de emisión de símbolos, hasta estabilizarse o estacionarse → **estado estacionario (V*)**

Ejemplo

	V0	V1	V2	V3		V9	V10
$p_0(t)$	1	1/2	3/8	11/32	...	0,333336	0,333334
$P_1(t)$	0	1/4	5/16	21/64	...	0,333332	0,333333
$p_2(t)$	0	1/4	5/16	21/64	...	0,333332	0,333333

Comprobación: V* = M.V*



Importante:
El estado estacionario es <u>independiente</u> de las condiciones iniciales

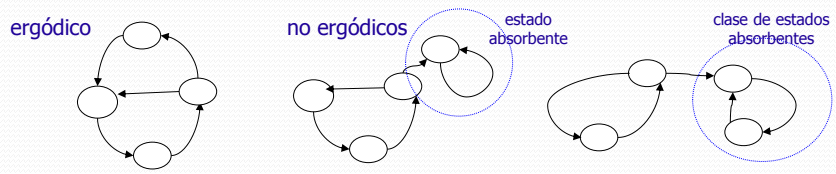
ESTADO ESTACIONARIO

- Como el estado estacionario es independiente de las condiciones iniciales puede obtenerse a partir de las probabilidades condicionales:

$$\mathbf{V^* = M \cdot V^*} \rightarrow \boxed{\begin{array}{l} \mathbf{(M - I)\, V^* = 0} \ (1) \\ \mathbf{\sum v_i^* = 1} \ (2) \end{array}}$$

Sistema de ecuaciones
(1) eliminar una de las ecuaciones
(2) Incorporar necesariamente

- Condiciones de existencia de V*:
  - ✓ conjunto finito de estados
  - ✓ fuente ergódica (todos los estados del proceso son alcanzables desde otro estado - no hay estados o clases absorbentes)



Ejemplo

$$M = \begin{array}{c|ccc|} & 0 & 1 & 2 \\ 0 & ½ & ¼ & ¼ \\ 1 & ¼ & ½ & ¼ \\ 2 & ¼ & ¼ & ½ \end{array}$$

$$\boxed{\begin{array}{l} (M - I)\, V^* = 0 \\ \sum v_i^* = 1 \quad (3') \end{array}}$$

$$M - I = \begin{array}{c|ccc|l} & 0 & 1 & 2 & \\ 0 & -½ & ¼ & ¼ & (1) \\ 1 & ¼ & -½ & ¼ & (2) \\ 2 & ¼ & ¼ & -½ & (3) \end{array}$$

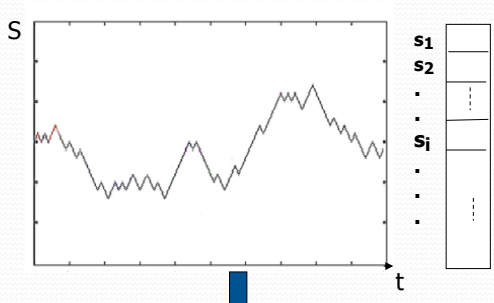
(1) - ½ v0* + ¼ v1* + ¼ v2* = 0
(2) ¼ v0* - ½ v1* + ¼ v2* = 0
(3) Se puede descartar
(3´) v0* + v1* + v2* = 1

Resolviendo el sistema de ecuaciones en este caso se obtiene : $V^* = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$

Nota: Para este ejemplo V* resulta con probabilidades de estado uniformes, por las características de M (pero no necesariamente es así en todos los casos!)

ESTADO ESTACIONARIO
por muestreo computacional

Cuando un proceso se encuentra en estado estacionario es más sencillo efectuar los cálculos por muestreo computacional



es suficiente simular una única trayectoria (o realización del proceso)

obtener las componentes de **V***:

$$\Rightarrow \mathbf{P(S^* = s_i)} \approx \frac{\#s_i}{\#\text{total transiciones}}$$

agregar más transiciones o pasos a la simulación hasta que no hayan cambios significativos en los valores calculados → **convergencia**

ESTADO ESTACIONARIO
por muestreo computacional

Para el ejemplo anterior:

$$M = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1/2 & 1/4 & 1/4 \\ 1 & 1/4 & 1/2 & 1/4 \\ 2 & 1/4 & 1/4 & 1/2 \end{array}$$

$$\text{Macum} = \begin{array}{c} 0 \\ 1 \\ 2 \end{array} \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 3/4 & 3/4 & 1/2 \\ 1 & 1 & 1 \end{pmatrix}$$

Sig_dado_Ant (col)
```
{r=rand()
  for(i=0 to 2)
    if ( r < Macum[i,col] )
      return i; }
```

condiciones iniciales (*según el problema*):

$$V_0 = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} \quad V0acum = \begin{pmatrix} 2/3 \\ 2/3 \\ 1 \end{pmatrix}$$

PrimerSimb ()
```
{ r=rand()
  for(i=0 to 2)
    if (r<V0acum[i])
      return i; }
```

converge (A[ ], B[])
```
{ for (i=0 to 2)
  { if (abs(A[i]-B[i]) > ε )
      return FALSE }
  return TRUE }
```

Calcular_Vector_estac
```
{
  emisiones=(0,0,0) //cant. de cada s_i
  V*= (0,0,0)   //vector est. actual
  V_ant=(-1,0,0)  //vector est. ant.
  pasos= 0

  s=PrimerSimb();

  while no converge (V*, V_ant)(1)
  { s=Sig_dado_Ant (s)
      pasos++;
      emisiones[s]++
      V_ant ← V*
      V* ← emisiones/#pasos
  } return V*  //estado estac.
}
```

(1) Se pueden agregar condiciones adicionales sobre cantidad de iteraciones

ENTROPÍA DE FUENTE MARKOVIANA

Si una fuente $S=\{s_1,s_2,\ldots, s_N\}$ se modeliza como una fuente con memoria (de orden 1) –o markoviana- mediante su matriz M de transición con probabilidades condicionales $\{p_{j/i}\}$ y posee probabilidades estacionarias $V^*=\{p_1^*,p_2^*,\ldots, p_N^*\}$

la entropía de la fuente markoviana S es:

$$\boxed{H_1(S) = \sum_i p_i * (-\sum_j p_{j/i} \log p_{j/i})}$$

probabilidades de emisión de los símbolos (del vector estacionario V*)

probabilidades condicionales de transición de símbolos (de la matriz de pasaje M)

Ejemplo

$$M = \begin{array}{c|ccc|} & 0 & 1 & 2 \\ 0 & ½ & ¼ & ¼ \\ 1 & ¼ & ½ & ¼ \\ 2 & ¼ & ¼ & ½ \end{array} \qquad V^* = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

H1(S)= [1/3 *(-½ $\log_2$ ½ - 2*¼ $\log_2$ ¼)]*3

=1,5 bits

BIBLIOGRAFÍA

Abramson N., Teoría de la Información y Codificación, Ed. Paraninfo, 1981

Cover T., Thomas J., Elements of Information Theory, 2nd ed., John Wiley & Sons, 2006

Papoulis A., Probability Random Variables and Stochastic Processes, McGraw-Hill, 1991

Shannon C., Weaver W., Teoría Matemática de la Comunicación, Ed.Forja, 1981