FACULTAD DE CIENCIAS EXACTAS
UNIVERSIDAD NACIONAL DEL CENTRO DE LA PROVINCIA DE BUENOS AIRES

TEORÍA DE LA INFORMACIÓN

Fuentes de Información (4)

INGENIERÍA DE SISTEMAS – 2022

Fuentes Markovianas

primera transición y primera recurrencia

Probabilidad de Transición en *n* pasos

(1º ecuación de Chapman-Kolmogorov)

$$p_{j/i}^{(n)} = \sum_{k} p_{k/i}^{(m)} \cdot p_{j/k}^{(n-m)} \quad \text{para } m<n$$

F → s_1 s_1 s_2 s_1 …

Probabilidad de **1°** Transición en *n* pasos

(2º ecuación de Chapman-Kolmogorov)

$$f_{j/i}^{(n)} = p_{j/i}^{(n)} - \sum_{m=1}^{n-1} f_{j/i}^{(m)} \cdot p_{j/j}^{(n-m)}$$

Media de **1°** Transición (si $F_{j/i} = 1$)

"tiempo medio de espera para pasar de i a j"

$$\mu_{j/i} = \sum_{n=1}^{\infty} n \cdot f_{j/i}(n)$$

Si la transición es de s_i al mismo s_i :

Probabilidad de **1°** Recurrencia en *n* pasos

$$f_{i/i}^{(n)} = p_{i/i}^{(n)} - \sum_{m=1}^{n-1} f_{i/i}^{(m)} \cdot p_{i/i}^{(n-m)}$$

Media de **1°** Recurrencia (si $F_{i/i} = 1$)

"tiempo medio de espera para retornar a s_i"

$$\mu_{i/i} = \sum_{n=1}^{\infty} n \cdot f_{i/i}^{(n)}$$

Probabilidad de 1° Recurrencia en *n* pasos

por muestreo computacional

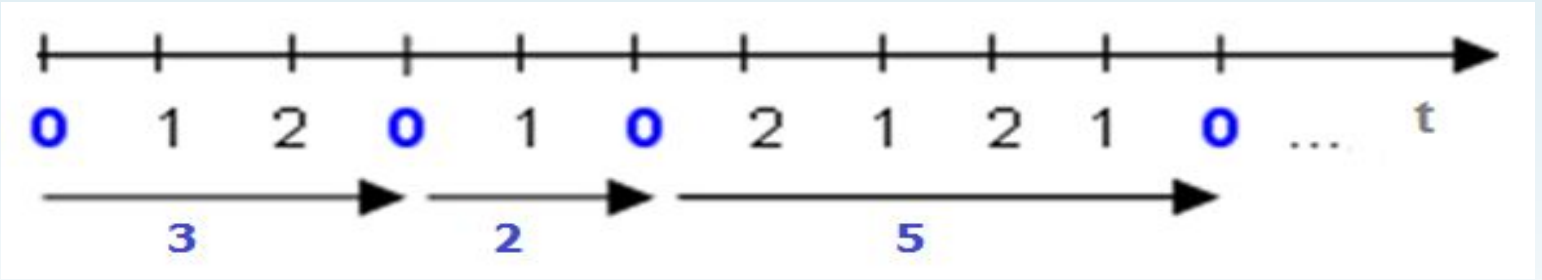
Probabilidad de volver a emitir por primera vez s_i en $t+n$, si se emitió s_i en t :

$$f_{i/i}^{(n)} = p_{i/i}^{(n)} - \sum_{m=1}^{n-1} f_{i/i}^{(m)} \cdot p_{i/i}^{(n-m)}$$

¿cómo obtenerla *por muestreo computacional*?

La simulación computacional no es "implementar la fórmula" usada para el cálculo analítico

Ejemplo: Obtener las distintas $f_{i/i}$ (n) para s_i=0



n= 2: $f_{0/0}^{(2)}$?

n= 3: $f_{0/0}^{(3)}$?

....

Probabilidad de 1° Recurrencia en *n* pasos

por muestreo computacional

Generar una secuencia de símbolos (mensaje) emitidos por la fuente según sus probabilidades y detectar la cantidad de pasos *n* que transcurren entre retornos sucesivos a s_i

$f_{i/i}(n)$ para s_i=0 y *n= 1, 2, 3 ….*

#retornos a s_i en *n* pasos

total de retornos a s_i

t_actual	símbolos generados aleatoriamente	retornos n= 1 2 3 …	total_retornos	$f_{i/i}$ n= 1 2 3 …	ult_ret
0	**0**	[0, 0, 0, …]	0	**[0, 0, 0, …]**	0
1	**1** (n=3)				
2	**2**				
3	**0**	[0, 0, 1, …]	1	**[0, 0, 1, …]**	3
4	**1** (n=2)				
5	**0**	[0, 1, 1, …]	2	**[0, ½, ½, …]**	5
…	…	…	…	…	

ult_ret → tiempo del último retorno a s_i

→ continuar iterando hasta convergencia del vector fi/i (*acotado*)

Probabilidad de 1° Recurrencia en *n* pasos

por muestreo computacional

Cálculo para un símbolo en particular

```
Prob_primera_recurrencia (simbolo)
{   retornos= [0, …, 0]    // #retornos a s_i en n pasos
    fi/i= [0, ..., 0]       //prob. primera recurrencia actual
    fi/i_ant= [-1, ..., -1] // prob. primera recurrencia anterior
    ult_ret= 0              // t del último retorno a simbolo
    t_actual= 0 ;  total_retornos= 0 ;
    s= simbolo     //parámetro (no es necesario generar Primer_Simbolo)
  while  not converge (fi/i, fi/i_ant) or (t_actual<T_MIN)
  {
      s= Sig_dado_Ant (s)
      t_actual++
      if ( s == simbolo )   // hay retorno
      {   n= t_actual - ult_ret
          retornos[n] ++
          total_retornos ++
          fi/i_ant ← fi/i
          fi/i ← retornos/total_retornos
          ult_ret= t_actual
      }
   }
   return fi/i
}
```

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

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

Probabilidad de 1° Recurrencia en *n* pasos

por muestreo computacional

Cómo plantearlo si se requiere calcular $\mathbf{f_{i/i}(n)}$ **para todos los** s_i **y** $n = 1, 2, 3, \ldots$?

Generar una secuencia de símbolos (mensaje) emitidos por la fuente según sus probabilidades y detectar la cantidad de pasos *n* que transcurren entre los retornos sucesivos de cada s_i

t_actual	símbolos generados aleatoriamente
0	**0**
1	**1**
2	**2**
⋮	⋮

total de retornos para cada s_i

0	0
1	0
2	0

total_retornos

#retornos en *n* pasos para cada s_i

n=	1	2	3	4	…
0	0	0	0	0	…
1	0	0	0	0	…
2	0	0	0	0	…

retornos

tiempo de última emisión para cada s_i

0	-1	0
1	-1	1
2	-1	2

ult_ret

Probabilidad de 1° Recurrencia en *n* pasos

por muestreo computacional

Cómo plantearlo si se requiere calcular $\mathbf{f_{i/i}(n)}$ **para todos los** s_i **y** $n = 1, 2, 3, \ldots$?

Generar una secuencia de símbolos (mensaje) emitidos por la fuente según sus probabilidades y detectar la cantidad de pasos *n* que transcurren entre los retornos sucesivos de cada s_i

t_actual	símbolos generados aleatoriamente
0	**0**
1	**1**
2	**2**
3	**0**
4	**2**
⋮	⋮

total de retornos para cada s_i

0	~~0~~	1
1	0	
2	~~0~~	1

total_retornos

#retornos en *n* pasos para cada s_i

n=	**1**	**2**	**3**	**4**	…
0	0	0	~~0~~1	0	…
1	0	0	0	0	…
2	0	~~0~~1	0	0	…

retornos

tiempo de última emisión para cada s_i

0	~~-1~~	~~0~~	3
1	~~-1~~	1	
2	~~-1~~	~~2~~	4

ult_ret

Probabilidad de 1° Recurrencia en *n* pasos

por muestreo computacional

Cómo plantearlo si se requiere calcular **$f_{i/i}(n)$ para todos los s_i y n= 1, 2, 3, ...** ?

Generar una secuencia de símbolos (mensaje) emitidos por la fuente según sus probabilidades y detectar la cantidad de pasos *n* que transcurren entre los retornos sucesivos de cada s_i

t_actual	símbolos generados aleatoriamente
0	**0**
1	**1**
2	**2**
3	**0**
4	**2**
5	**1**
⋮	⋮

total de retornos para cada s_i

0	~~0~~	1
1	~~0~~	1
2	~~0~~	1

total_retornos

#retornos en *n* pasos para cada s_i

n=	**1**	**2**	**3**	**4**	...
0	0	0	~~0~~ 1	0	...
1	0	0	0	~~0~~ 1	...
2	0	~~0~~ 1	0	0	...

retornos

tiempo de última emisión para cada s_i

0	~~-1~~	~~0~~	3
1	~~-1~~	~~1~~	5
2	~~-1~~	~~2~~	4

ult_ret

→ Calcular $f_{i/i}^{(n)}$= retornos(s_i,n) /total_retornos(s_i) y verificar si converge

Media de 1° Recurrencia

por muestreo computacional

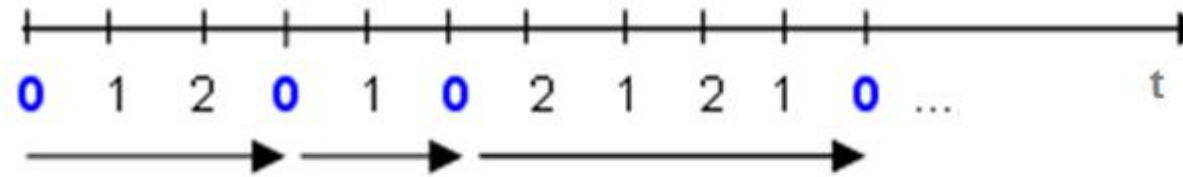
$$\mu_{i/i} = \sum_{n=1}^{\infty} n \cdot f_{i/i}(n)$$

tiempo medio de espera para retornar a s_i

No se trata de "implementar la fórmula" del cálculo analítico

Generar una secuencia de símbolos (mensaje) emitidos por la fuente según sus probabilidades y registrar el promedio de pasos entre retornos sucesivos a s_i

Ejemplo:
Cálculo de $\mu_{0/0}$



suma de #pasos de retorno a 0 = 3 + 2 + 5 + ... = t_actual

retornos a 0 = 1 2 3

$$\mu_{0/0} = \frac{\text{suma de \#pasos de retorno a 0}}{\text{\# retornos a 0}} = \frac{\text{t\_actual}}{\text{\# retornos a 0}}$$

Media de 1° Recurrencia

por muestreo computacional

Cálculo para un símbolo en particular

```
Media_recurrencia (simbolo)
{   retornos= 0        // #retornos a s_i
    media= 0           //media recurrencia actual
    media_ant= -1      //media recurrencia anterior
    t_actual= 0
    s= simbolo         //parámetro (no es necesario generar Primer_Simbolo)
  while  not converge (media, media_ant) or (t_actual<T_MIN)
  {
      s= Sig_dado_Ant (s)
      t_actual++
      if ( s == simbolo )    // hay retorno
      {
          retornos++
          media_ant= media
          media= t-actual/retornos
       }
   }
   return media
 }
```

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

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

```
CalcularProbabilidadSumaDados(int suma)

os = 0;
das = 0;
rob = 0;
robAnterior = -1;

while (!this.Converge(prob, probAnterior))
{
    int dado1 = this.ArrojarDado();
    int dado2 = this.ArrojarDado();

    if (dado1 + dado2 == suma) {
        exitos++;
    }
    tiradas++;

    probAnterior = prob;
    prob = (float) (exitos) / tir
}

return prob;
```