Algoritmos aleatorizados: Cadenas de Markov

J. M. Almira

Universidad de Murcia

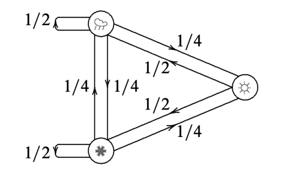
Parte 1.

Concepto y Primeros ejemplos de Cadenas de Markov

El tiempo en Salzburgo

Los turistas italianos son muy aficionados a Salzburgo, la “Roma del Norte”. Al llegar allí, descubren rápidamente que el clima no es tan estable como en el sur. Allí nunca hay dos días consecutivos de buen tiempo. Si un día es soleado, entonces el día siguiente llueve o nieva con igual probabilidad. Un día lluvioso o con nieve es seguido con igual probabilidad por un día con el mismo tiempo o por un cambio de tiempo. Y en caso de que el tiempo cambie, mejora sólo en la mitad de los casos. Denotemos los tres posibles estados del tiempo en Salzburgo como soleado, lluvioso y nevado. La siguiente matriz y la correspondiente figura ilustran la situación:

$$\begin{array}{c} \\ \text{☼} \\ \text{☂} \\ \text{❄} \end{array} \begin{array}{c} \begin{array}{ccc} \text{☼} & \text{☂} & \text{❄} \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$



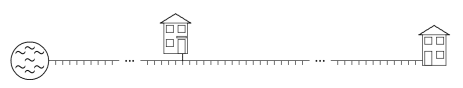
El tiempo en Salzburgo

Preguntas:

- Llueve hoy. ¿Cuál es la probabilidad de que dentro de dos semanas amanezca soleado?
- ¿Cuántos días de lluvia esperamos (en promedio) durante el próximo mes?
- ¿Cuál es la duración media de un período de mal tiempo?

El paseo del borracho

Un borracho quiere volver a casa desde el pub. El pub está situado en una calle recta. A la izquierda, después de 100 pasos, termina en un lago, mientras que la casa del borracho está a la derecha a una distancia de 200 pasos del pub. En cada paso, el borracho camina hacia su casa (con probabilidad $2/3$) o hacia el lago (con probabilidad $1/3$). Si llega al lago, se ahoga. Si vuelve a casa, se queda y se va a dormir.



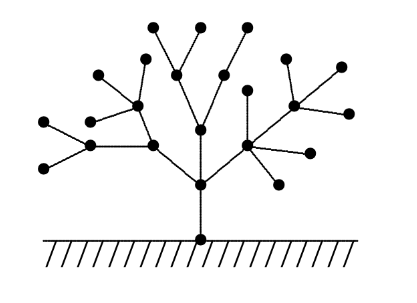
El paseo del borracho

Preguntas:

- ¿Cuál es la probabilidad de que el borracho se ahogue? ¿Cuál es la probabilidad de que regrese a casa?
- Suponiendo que logra volver a casa, ¿cuánto tiempo (¿cuántos pasos?) tarda en promedio?

El gato y el árbol

Un gato sube a un árbol.



En cada punto de ramificación, decide al azar, con la misma probabilidad entre todas las posibilidades, volver a bajar al punto de ramificación anterior o avanzar a uno de los nodos vecinos más altos. Si el gato llega a la parte superior (a una "hoja"), en el siguiente paso volverá al punto de ramificación anterior y continuará como antes.

El gato y el árbol

Preguntas:

- ¿Cuál es la probabilidad de que el gato regrese al suelo? ¿Cuál es la probabilidad de que regrese en el paso n-ésimo?
- ¿Cuánto tiempo tarda en promedio en regresar al suelo?
- ¿Cuál es la probabilidad de que el gato regrese al suelo antes de visitar cualquier hoja (o una determinada) del árbol?
- ¿Con qué frecuencia visitará el gato, en promedio, una “hoja” dada antes de regresar al suelo?

Definición (Cadena de Markov (Finita))

Llamamos cadena de Markov a toda sucesión de variables aleatorias $\{X_n\}_{n=0}^{\infty}$ *que cumple los siguientes requisitos:*

- *Todas las v.a.* X_n *comparten el mismo rango, que es un conjunto finito* $S = \{s_1, \cdots, s_k\}$ *al que denominamos espacio de estados de la cadena. Los elementos de* S *se llaman estados de la cadena.*
- *Existe una matriz* $P = (p_{i,j})_{i,j=1}^{k}$, *llamada matriz de transiciones, tal que*

$$\begin{aligned} &P(X_{n+1} = s_j | X_0 = s_{i_0}, \cdots, X_{n-1} = s_{i_{n-1}}, X_n = s_i) \\ &= P(X_{n+1} = s_j | X_n = s_i) = p_{i,j};\ 1 \leq i,j \leq k; n = 0, 1, 2, \cdots . \end{aligned}$$

Esto significa que el estado futuro de la cadena (X_{n+1}) *solo depende de su estado presente* (X_n) *y no de su historia pasada (los valores* $X_0, X_1, X_2, \cdots, X_{n-1}$*).*

Teorema (Distribución de Probabilidades de X_n)

Sea (X_n) una cadena de Markov finita con matriz de transición $P = (p_{i,j})$. Si

$$\mu^{(0)} = (\mu_1^{(0)}, \cdots, \mu_k^{(0)})$$

es la distribución de probabilidades inicial (i.e., de X_0), entonces, para cada $n \in \mathbb{N}$, la distribución de probabilidades de X_n viene dada por

$$\mu^{(n)} = \mu^{(0)} P^n$$

Notación: Si $\mu = (\mu_1, \cdots, \mu_m)$ y $A = (a_{ij})$, usamos la notación:

$$(\mu)_j = \mu_j \text{ y } (A)_{ij} = a_{ij}$$

Demostración. Asumimos $S = \{1, \cdots, k\}$. Para $n = 0$ es verdad porque $P^0 = I$. Supongamos que el resultado es cierto para n y consideremos

$$\mu^{(n+1)} = (P(X_{n+1} = 1), \cdots, P(X_{n+1} = k))$$

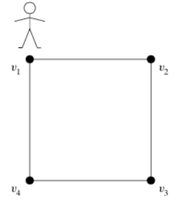
la distribución de probabilidades de X_{n+1}. Entonces

$$\begin{aligned}
\mu_j^{(n+1)} &= P(X_{n+1} = j) = \sum_{i=1}^{k} P(X_{n+1} = j | X_n = i) P(X_n = i) \\
&= \sum_{i=1}^{k} \mu_i^{(n)} P(X_{n+1} = j | X_n = i) \\
&= \sum_{i=1}^{k} \mu_i^{(n)} p_{ij} \\
&= (\mu^{(n)} P)_j \text{ aplicamos ahora la hipótesis de inducción, } \mu^{(n)} = \mu^{(0)} P^n \\
&= (\mu^{(0)} P^n P)_j = (\mu^{(0)} P^{n+1})_j \text{ para } j = 1, \cdots, k.
\end{aligned}$$

Esto finaliza la prueba.

Un primer ejemplo: caminos aleatorios en grafos

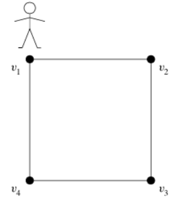
Consideremos el grafo de la figura.



Sobre él colocamos un caminante que va a seguir un paseo aleatorio, de vértice en vértice, cambiando de posición cada segundo. Denotamos por X_n el índice del vértice del grafo ocupado por nuestro caminante en el instante n. Imponemos $X_0 = 1$. A partir de la posición v_i (en cualquier instante n), el caminante saltará (en el instante $n+1$) con probabilidad $1/2$ a cualquiera de los vértices adyacentes a v_i (y con probabilidad 0 a los vértices no adyacentes a su posición en el instante n)

El ejemplo más básico: caminos aleatorios en grafos

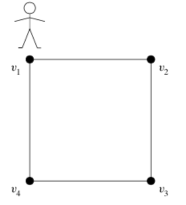
Consideremos el grafo de la figura.



En nuestro caso, la matriz de transición $P = (p_{i,j})$, donde $p_{i,j} = P(X_{n+1} = v_j | X_n = v_i)$, viene dada por:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

El ejemplo más básico: caminos aleatorios en grafos



Un simple cálculo nos indica que

$$P^2 = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} ; P^3 = P$$

Se sigue que $P^{2k} = P^2$ y $P^{2k+1} = P$, para todo k.

El ejemplo más básico: caminos aleatorios en grafos

Por tanto, la distribución de probabilidad de X_n oscila entre

$$\mu^{(0)}P^{2k+1} = (1,0,0,0)\begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix} = (0,\frac{1}{2},0,\frac{1}{2})$$

y

$$\mu^{(0)}P^{2k} = (1,0,0,0)\begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} = (\frac{1}{2},0,\frac{1}{2},0)$$

según $n \in \{2k+1, 2k\}$ sea impar o par.

Una nota sobre Potencias de Matrices y Diagonalización de Matrices:

- Sea A matriz cuadrada. Si la descomponemos como

$$A = Q^{-1}DQ$$

con D matriz diagonal (es decir, si A es diagonalizable), entonces

$$A^n = Q^{-1}D^nQ,$$

que es muy fácil de calcular.

Teorema

Una matriz $A \in M_k(\mathbb{R})$ es diagonalizable si y solo si existe una base de $\mathbb{R}^k$ formada por autovectores de A, lo cual es, a su vez, equivalente a que la suma de las dimensiones de los autoespacios de A valga k.

Teorema

Toda matriz simétrica es diagonalizable (con una base ortogonal)

De vuelta al ejemplo de los caminos aleatorios en grafos

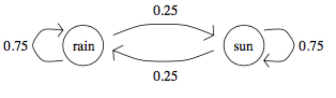
Podemos aplicar lo anterior a nuestro ejemplo, pues $P = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$ es simétrica. Sus autovalores son $\{-1, 1, 0\}$ y la descomposición queda:

$$P = \begin{bmatrix} -1 & 1 & 0 & -1 \\ 1 & 1 & -1 & 0 \\ -1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

Se sigue que $P^{2k} = P^2$ y $P^{2k+1} = P$, para todo n, por lo que la distribución de probabilidad de X_n solo depende de la paridad de n, tal como vimos antes.

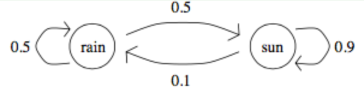
Predicción del tiempo (otra vez)

Consideramos dos casos de predicción del tiempo, pues la matriz de transición de probabilidades para los estados de lluvia o sol, dependen de la ubicación de la localidad en la que se va a realizar la predicción. Ejemplo 1: Una localidad del norte de España.



Aquí la matriz de transición es $P = \begin{bmatrix} 0{,}75 & 0{,}25 \\ 0{,}25 & 0{,}75 \end{bmatrix}$

Ejemplo 2: Una localidad del sur de España.



Aquí la matriz de transición es $P = \begin{bmatrix} 0{,}5 & 0{,}5 \\ 0{,}1 & 0{,}9 \end{bmatrix}$

Otro ejemplo: Surfear por la World Wide Web

Si denotamos por $S = \{s_k\}_{k\in\Delta}$ las páginas webs que existen en Internet en un momento dado (y asumimos que dicho número permanece constante para nuestros cálculos) y por X_n la v.a. que nos dice en qué página de la red estamos en el instante n, y asumimos que viajamos a través de la red de forma aleatoria, de modo que en cada momento nos movemos desde la página s_i que ocupamos eligiendo aleatoriamente entre aquellas que están enlazadas a esta, tendremos una Cadena de Markov gigantesca cuya matriz de transición es $P = (p_{ij})$, siendo

$$p_{ij} = \begin{cases} \frac{1}{d_i} & \text{si } s_j \text{ está enlazada a } s_i \\ 0 & \text{en otro caso} \end{cases}$$

Aquí $d_i =$ número de enlaces con s_i.

Parte 2.

Simulación de Cadenas de Markov

Cómo simular Cadenas de Markov (X_n)

Hipótesis inicial: Sabemos generar una m.a.s. de $U \sim \mathbf{U}([0,1])$

- La generación de sucesiones de números aleatorios es un problema complicado.
- Suponemos que en efecto disponemos de una m.a.s. (U_n) de la v.a. uniforme con rango $[0,1]$

Nota: Ver el Capítulo 17 de: W.J. Stewart, Probability, Markov Chains, Queues, and Simulation. Princeton University Press. 2009.
También puede verse el Tomo 2 de D.E. Knuth "The art of Programing Computers"(Dedicado a algoritmos seminuméricos).

Problema: Dados P y $\mu^{(0)}$, ¿es posible simular la cadena de Markov asociada (X_n)?

La respuesta es: Sí!

$$\begin{cases} X_0 & = & \Phi(U_0) \\ X_{n+1} & = & \Psi(X_n, U_{n+1}) \end{cases}$$

Para hacerlo necesitamos ($S = \{s_1, \cdots, s_k\}$ denota el espacio de estados de la CM):

- Una función de inicialización $\Phi : [0,1] \to S$ que proporciona el valor de X_0 de acuerdo con $\mu^{(0)}$
- Una función de actualización $\Psi : S \times [0,1] \to S$ que genera el resto de valores de la Cadena de Markov.

Construcción de la función de inicialización $\Phi : [0, 1] \to S$

Proposición

Supongamos que $U_0 \sim \mathbf{U}([0, 1])$ *y que* $\Phi : [0, 1] \to S$ *es tal que*

$$\int_0^1 \chi_{\{\Phi(x)=s\}}(x)dx = \mu^{(0)}(s) \text{ para todo } s \in S.$$

Entonces $X_0 = \Phi(U_0)$ *sigue una distribución cuya fdp es* $\mu^{(0)}$.

Construcción de la función de inicialización $\Phi : [0, 1] \to S$

Proposición

Supongamos que $U_0 \sim \mathbf{U}([0, 1])$ *y que* $\Phi : [0, 1] \to S$ *es tal que*

$$\int_0^1 \chi_{\{\Phi(x)=s\}}(x)dx = \mu^{(0)}(s) \text{ para todo } s \in S.$$

Entonces $X_0 = \Phi(U_0)$ *sigue una distribución cuya fdp es* $\mu^{(0)}$.

Demostración: En efecto,

$$P[X_0 = s] = P[\Phi(U_0) = s] = \int_0^1 \chi_{\{\Phi(x)=s\}}(x)dx = \mu^{(0)}(s) \text{ para todo } s \in S$$

Construcción de la función de inicialización $\Phi : [0,1] \to S$

Proposición

Supongamos que $U_0 \sim \mathbf{U}([0,1])$ *y que* $\Phi : [0,1] \to S$ *es tal que*

$$\int_0^1 \chi_{\{\Phi(x)=s\}}(x)dx = \mu^{(0)}(s) \text{ para todo } s \in S.$$

Entonces $X_0 = \Phi(U_0)$ *sigue una distribución cuya fdp es* $\mu^{(0)}$.

Por tanto, buscamos $\Phi : [0,1] \to S$ tal que

$$\int_0^1 \chi_{\{\Phi(x)=s\}}(x)dx = \mu^{(0)}(s)$$

para todo $s \in S$.

Construcción de la función de inicialización $\Phi : [0, 1] \to S$

Por tanto, buscamos $\Phi : [0, 1] \to S$ tal que

$$\int_0^1 \chi_{\{\Phi(x)=s\}}(x)dx = \mu^{(0)}(s)$$

para todo $s \in S$. Basta tomar

$$\Phi(x) = s_i \text{ sii } x \in \left[\sum_{j=1}^{i-1} \mu^{(0)}(s_j), \sum_{j=1}^{i} \mu^{(0)}(s_j)\right)$$

Construcción de la función de actualización $\Psi : S \times [0,1] \to S$

Teorema

Sea (U_n) sucesión de v.a. indep., id. dist. con distribución $U([0,1])$. Sea P la matriz de transición de una CM con $k = \#S$ estados y sea Φ una función de inicialización. Supongamos que $\Psi : S \times [0,1] \to S$ satisface que, para cada i,

$$\int_0^1 \chi_{\{\Psi(s_i,x)=s_j\}}(x)dx = P_{i,j}.$$

Entonces

$$\begin{cases} X_0 & = & \Phi(U_0) \\ X_{n+1} & = & \Psi(X_n, U_{n+1}) \end{cases}$$

define una Cadena de Markov (X_n) con estados en S, matriz de transición P, y distribución inicial $\mu^{(0)}$.

Construcción de la función de actualización $\Psi : S \times [0,1] \to S$

Demostración. En efecto:

$$\begin{aligned}
P[X_{n+1} = s_j | X_n = s_i] &= P[\Psi(X_n, U_{n+1}) = s_j | X_n = s_i] \\
&= P[\Psi(s_i, U_{n+1}) = s_j | X_n = s_i] \\
&= P[\Psi(s_i, U_{n+1}) = s_j] \\
& \quad \text{(pues } U_{n+1} \text{indep. de } X_k \text{ para } k \leq n\text{)} \\
&= \int_0^1 \chi_{\{\Psi(s_i,x)=s_j\}}(x)dx \\
&= P_{i,j}.
\end{aligned}$$

Construcción de la función de actualización $\Psi : S \times [0,1] \to S$

La función Ψ se construye de forma análoga a lo que se hizo con Φ:

$$\Psi(s_i, x) = s_j \text{ sii } x \in \left[\sum_{h=1}^{j-1} P_{ih}, \sum_{h=1}^{j} P_{ih}\right)$$

Un ejemplo hecho en SAGE

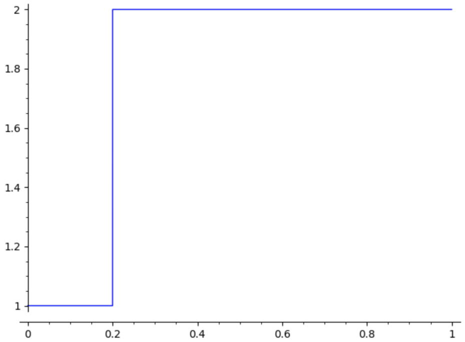
```
P = matrix(CC ,[[0.175 , 0.825],[0.526 ,0.474]])
P
```

```
[0.175000000000000 0.825000000000000]
[0.526000000000000 0.474000000000000]
```

```
Phi=piecewise([[(0,0.2),1],[(0.2,1),2]]);

Phi

plot(Phi(x),(x,0,1))
```



Un ejemplo hecho en SAGE

```
def psi(n,x):
    if n==1 and 0<=x<0.175:
        return 1
    if n==1 and 0.175<=x<=1:
        return 2
    if n==2 and 0<=x<0.526:
        return 1
    if n==2 and 0.526<=x<=1:
        return 2
```

```
psi(2,0.1)
```

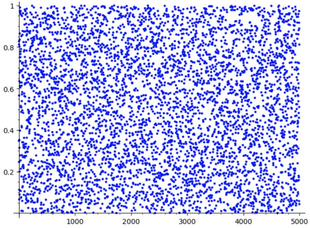
```
    1
```

Un ejemplo hecho en SAGE

```
import numpy as np
U=np.random.uniform(0,1,50000)
U[0:100]
```

```
    array([0.40083746, 0.34835457, 0.82791346, 0.81265363, 0.24592586,
           0.56637186, 0.06589405, 0.11238956, 0.35229126, 0.63509451,
           0.31457454, 0.80275195, 0.67648502, 0.08938754, 0.19994731,
           0.8358177 , 0.8651584 , 0.97309107, 0.61223356, 0.27768768,
           0.39760514, 0.99053884, 0.62899497, 0.96493988, 0.97074842,
           0.73180532, 0.60526306, 0.75091456, 0.7666011 , 0.68757348,
           0.77862637, 0.25376806, 0.97411035, 0.10658911, 0.51654766,
           0.21372431, 0.51079361, 0.9564458 , 0.79047488, 0.07380427,
           0.68786802, 0.07634282, 0.48728061, 0.68317815, 0.50951518,
           0.85263426, 0.17623457, 0.01099148, 0.61374985, 0.41928304,
           0.62900189, 0.68700131, 0.41304873, 0.53957767, 0.27905101,
           0.57188803, 0.94960847, 0.72396035, 0.7202713 , 0.58304391,
           0.34136857, 0.05873661, 0.49491693, 0.12407519, 0.85878075,
           0.22377223, 0.87735199, 0.65901561, 0.66421094, 0.12917705,
           0.00957544, 0.26947027, 0.72470909, 0.69390444, 0.96242346,
           0.23199632, 0.48313533, 0.34083276, 0.23007376, 0.6540411 ,
           0.32836486, 0.31655614, 0.22455553, 0.82357648, 0.21678286,
           0.16214261, 0.32728472, 0.17354589, 0.71079305, 0.08778531,
           0.15032938, 0.77877311, 0.87461913, 0.36241578, 0.57623361,
           0.80728739, 0.27496268, 0.31287885, 0.7851691 , 0.56392331])
```

```
list_plot(U[1:5000])
```



Un ejemplo hecho en SAGE

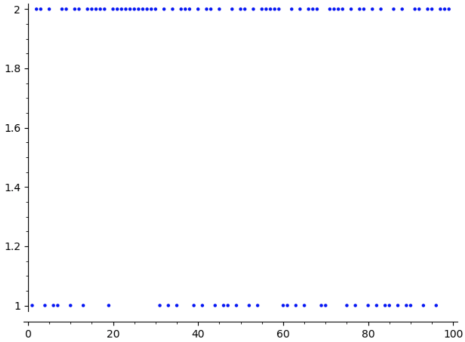
```
def X(n):
    if n==0:
        return Phi(U[0])
    else:
        return psi(X(n-1),U[n])
```

```
[X(0),X(1),X(2),X(3),X(50)]
```

```
[2, 1, 2, 2, 2]
```

Un ejemplo hecho en SAGE

```
list_plot([(k,X(k)) for k in range(1,100)])
```



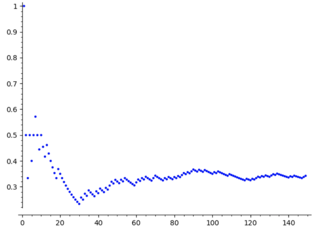
Un ejemplo hecho en SAGE

```
def Y(n):
    if X(n)==1:
        return 1
    else:
        return 0
```

```
def S(n):
    if n==0:
        return Y(0)
    else:
        return Y(n)+S(n-1)
```

Un ejemplo hecho en SAGE

```
list_plot([(k,S(k)/k) for k in range(1,150)])
```



```
list_plot([(k,S(k)/k) for k in range(1,250)])
```