

5. Autómatas celulares y simulación de procesos en el espacio (1): principios

Modelización de Sistemas Ambientales (07M1)
Dpto. de Ecología e Hidrología y Dpto. Geografía
Facultad de Biología
Universidad de Murcia

Índice

Curso 2006–07

1. Introducción	1
2. Movimiento brauniano (2 horas)	1
3. Para entregar	4

1. Introducción

Esta práctica está destinada a estudio de la simulación con la ayuda de autómatas celulares. Son estos objetos virtuales que presentan un estado asociado a una serie de propiedades, descritas por un conjunto de variables, y que cambia de estado atendiendo a un colección de reglas prefijadas. Por ejemplo, podemos simular individuos de una población humana para los que se describe el estado de infección con respecto a una enfermedad con una variable. Cada unidad de tiempo se somete a los individuos a un "sorteo" para determinar cuales han sido infectados, cuales han sanado y cuales mantienen su estado de salud.

Nos interesa especialmente los casos en los que el espacio juega un papel importante. En el ejemplo anterior el mecanismo de infección puede asociarse a un contagio debido a la proximidad a un individuo enfermo.

La utilización de autómatas celulares es una herramienta auxiliar de muchas ciencias y en muchos casos exige una gran capacidad de computación.

2. Movimiento brauniano (2 horas)

Un caso sencillo puede estudiarse con el movimiento de partículas considerando un comportamiento puramente brauniano, o browniano. Los objetos son partículas descritas por las coordenadas de su posición (x, y); las reglas: cada partícula tendrá un desplazamiento l , que sigue una $\mathcal{N}(0, 1)$. Además, la dirección, d , también presenta un comportamiento aleatorio, que para un una espacio bidimensional sigue una distribución uniforme de 0 a 2π : $\mathcal{U}(0, 2\pi)$.

EJERCICIOS

1. Considerando una partícula puede calcularse su movimiento mediante:

```
# Posición original
x<-y<-0 #creamos x e y
x<-0
y<-0

#Nueva dirección
d<-runif(1,0,2*pi)

#Desplazamiento
l<-rnorm(1,0,1)
```

```

#Variación de coordenadas
ix<-l*cos(d)
iy<-l*sin(d)

#Nuevas coordenadas
x<-x+ix
y<-y+iy

#Representación
plot(x,y, xlim=c(-2,2), ylim=c(-2,2))
text(x,y,"Posición final",pos=1)
points(0,0)
text(0,0,"Posición original",pos=1)

```

¿Realizar un esquema del movimiento de la partícula representando los distintos factores que influyen en él?

¿Cómo puede calcularse la distancia se aleja la partícula de su posición inicial?

2. Para trabajar más fácilmente con autómatas se puede describir el cambio de posición de las partículas mediante un vector: $p \leftarrow c(x, y)$. La siguiente función, `mb()` permite simplificar el manejo del autómata:

```

mb<-function(p=c(0,0)) {
# el argumento 'p' es la posición inicial
# de la partícula. Si no se especifica es el origen
# de coordenadas.

d<-runif(1,0,2*pi)
l<-rnorm(1,0,1)
ix<-l*cos(d)
iy<-l*sin(d)
(mb<-c(ix,iy))
}

```

Utilizando el siguiente protocolo se puede representar un autómata desplazado braunianamente

```

p<-c(0,0) #posición inicial
q<-mb(p) #posición final
plot(q[1],q[2], xlim=c(-2,2), ylim=c(-2,2))
points(p[1],p[2])

```

¿Muestra el gráfico lo esperado? ¿Y al repetirlo?

3. Con el siguiente protocolo comprobaremos el comportamiento de una muestra con un gran número de partículas. El tamaño de la muestra puede modificarse cambiando el valor de `np` para aumentar el número de partículas en juego. La simulación puede tardar mucho si tenemos muchas partículas, paciencia.

```

np<-1000
plot(0,0,xlim=c(-3,3),ylim=c(-3,3))
for(i in 1:np){
  p<-mb()
  points(p[1],p[2])
}

```

¿Qué aspecto tiene el gráfico de puntos obtenido? ¿Se corresponde con lo esperado?

4. Con el fin de tener otra visión de los resultados de la simulación anterior vamos a tabular la posición de los datos y representar la densidad de puntos por unidad de superficie.

En primer lugar, generaremos una población de autómatas cuya localización estará descrita en la matriz `pob`:

```
np<-100000
pob<-matrix(0,np,2) #todos los individuos en 0,0
for (i in 1:np){
  pob[i,]<-mb(pob[i,])
}
```

Ahora tabulando los datos:

```
tabla <- table(round(pob[,1],1),round(pob[,2],1))
tabla # de dimensiones poco presentables en pantalla
```

```
image(tabla) #la tabla vista como un mapa
```

```
niveles<-c(10,25,50,250,100,250,500)
contour(tabla,levels=niveles,add=T) #isopletras
```

¿Puede afirmarse que la imagen muestra un comportamiento de los autómatas isotrópico?

- Utilizaremos, ahora, en lugar de las frecuencias observadas en la tabla sus logaritmos, con el fin de evitar el efecto extremo de la distribución:

```
image(log10(tabla))
contour(log10(tabla),levels=log10(niveles),add=T)
```

¿Qué ventajas tiene esta representación? ¿Podría mejorarse para una lectura mas sencilla?

- Los resultados anteriores pueden compararse con una representación tridimensional de la tabla, considerando la frecuencia como el valor a representar en el eje z :

```
persp(as.matrix(tabla),zlim=c(0,10000))
```

En este caso la visualización mejora truncando el eje z para el valor $np \cdot 0.005$

```
persp(as.matrix(tabla),zlim=c(0,np*0.005))
```

Considerando las variables *posición del autómata en el eje*, tanto para abscisas como para ordenadas ¿Sugiere el gráfico la distribución normal¹?

- El comportamiento de las variables, mencionadas anteriormente, en cada uno de los ejes puede verse mediante la representación de la suma de las frecuencias marginales, tanto para las filas:

```
valores.y <-as.numeric(rownames(tabla))
marginales.fil<-apply(tabla,1,sum)
plot(valores.y,marginales.fil,type="h")
```

como para las columnas:

```
valores.x <-as.numeric(colnames(tabla))
marginales.col<-apply(tabla,2,sum)
plot(valores.x,marginales.col,type="h")
```

¿Se trata de un comportamiento normal? ¿Tiene alguna característica en común estas distribuciones con la normal?

¹Para obtener una distribución conjunta de dos variables bastará con son simularla:

- x e y normales:
table(round(rnorm(1000000),1),round(rnorm(1000000),1))
- x normal e y uniforme:
table(round(rnorm(1000000),1),round(runif(1000000),1))
- ...

8. Consideraremos ahora el comportamiento de la variable *distancia de la posición actual con respecto a la original*. Podemos calcular las distancias y estudiar su distribución de frecuencias mediante:

```
distancia<-sqrt (pob[,1]^2+pob[,2]^2)
hist (distancia,br=seq(0,5,0.1))->histx
```

¿Se trata de un comportamiento normal?

9. Utilizaremos los valores de tabulación que proporciona la función `hist` para representar conjuntamente la distribución obtenida y la correspondiente a la normal. Estos valores son `histx$mids` y `histx$den`, respectivamente, los puntos medios de cada intervalo y las densidades (frecuencia) asociada a cada uno de ellos:

```
plot (histx$mids, histx$den, type="h")
points (histx$mids, dnorm (histx$mids))
```

¿Coinciden los valores obtenidos con los teóricos?

```
plot (histx$mids, histx$den/2, type="h")
points (histx$mids, dnorm (histx$mids))
```

¿Y ahora? ¿Cuál es la diferencia entre este procedimiento y el anterior? ¿Sigue una distribución normal la variable *distancia al origen*? ¿Cuál es la causa que nos lleva a introducir $\frac{1}{2}$ en la expresión?

¿A qué distancia de la situación original está al menos la mitad de las partículas? ¿Y el 90% de ellas?

¿Podrían haberse obtenido estos datos directamente de la tabla de la distribución normal obtenida en la práctica 1? ¿Cómo?

¿Podemos hablar de “una” o más distancia de desplazamiento? ¿Qué opciones tenemos?

3. Para entregar

¿Qué puede decirse del comportamiento de la variable *distancia a la posición de origen* si la función `mb()` en lugar de considerar desplazamientos normales lo hiciese mediante un desplazamiento uniforme?

Para discutirlo utilizar el esquema habitual (Introducción, métodos, resultados, ...), considerando sólo los apartados que resulten necesarios.

```
mb<-function(p=c(0,0)) {
# el argumento 'p' es la posición inicial
# de la partícula. Si no se especifica es el origen
# de coordenadas.
```

```
d<-runif(1,0,2*pi)
l<-runif(1)
ix<-l*cos(d)
iy<-l*sin(d)
(mb<-c(ix,iy))
}
```