

Descripción estadística y procesos de transporte en un medio aleatorio

Los medios aleatorios (*random media*) tienen en física muchas aplicaciones. Desde la descripción del flujo a través de medios porosos hasta la conducción electrónica en medios amorfos (no cristalinos). Para describir la estructura de un medio aleatorio es necesario conocer su estructura de “huecos”, esto es, de zonas del poroso que ofrecen facilidad a los procesos de transporte. Un modelo muy simplificado, pero ilustrativo, de medio aleatorio es el “poroso unidimensional”, cuyas propiedades estudiaremos a continuación.

1. Consideremos el medio como una cadena de sitios que pueden estar ocupados o no ocupados (un array de valores 0 = “no ocupado” y 1 = “ocupado”). Estos sitios están dispuestos aleatoriamente, esto es, si el sitio i -ésimo tiene un valor 0, la probabilidad de que el $(i + 1)$ -ésimo siguiente tenga un valor 0 o 1 es independiente del anterior. El cociente entre el número de 0s y 1s, se denomina porosidad del medio.

Ejercicio 1: Simular, usando un generador de números aleatorios, un poroso unidimensional de longitud L y porosidad r introducidas por el usuario. Guárdense los valores sucesivos de 0 o 1 en un archivo de texto y represéntense con Gnuplot para ver el aspecto del poroso. Úsense tamaños $L = 250, 1250$ y porosidades $r = 0,25, 1, 4$, mostrando los resultados en una tabla de 2×3 gráficas.

Nota: Nótese que la probabilidad de que un sitio de la cadena esté ocupado viene dada por $p = 1/(1 + r)$.

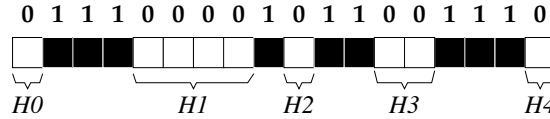


Figura 1: Poroso 1-D y huecos en él.

2. Consideremos la distribución estadística del tamaño de los huecos en una cadena de longitud L . La longitud de un hueco es el número de 0s consecutivos que lo forman. Interesa saber cuántos huecos de tamaño l hay por unidad de longitud de la cadena. O, dicho de otro modo, saber cuál es la probabilidad de que, si se escoge un 0 al azar en la cadena, éste pertenezca a un hueco de longitud l .

Ejercicio 2: Calcular el histograma del número h de huecos de distintos tamaños l . Genérense N cadenas de longitud L y porosidad r . Calcúlense los histogramas de los huecos de tamaños $1, 2, \dots, L/2$ normalizando las frecuencias con el producto $N \times L$. Compruébese

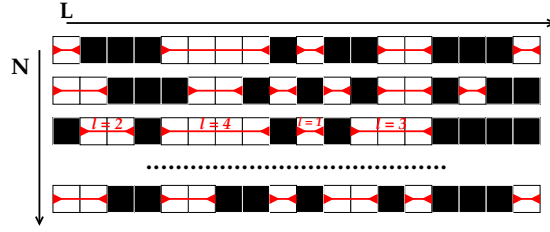


Figura 2: Matriz de poros y definición de longitud de correlación.

gráficamente que ese histograma viene aproximadamente dado por

$$h(l, r) = r^l / (1 + r)^{2+l}$$

Pruébese con tamaños $L = 64, 1024$ y porosidades $r = 0,25, 1, 4$.
¿Es igualmente bueno el ajuste del histograma para ambos tamaños?

3. El histograma de longitudes de huecos es una descripción muy detallada del medio aleatorio con el que estamos trabajando. No obstante, lo que suele interesar en el estudio de los fenómenos de transporte son valores medios de magnitudes que caracterizan al medio (como su porosidad r) o a los fenómenos de transporte que suceden en él. En la cadena, por ejemplo, interesa saber hasta qué distancia, en promedio, podrá alcanzar un fenómeno de transporte si lo iniciamos en un 0 escogido al azar, esto es, partiendo de ese 0, cuántos sitios de la cadena podrá recorrer (hacia un lado o hacia otro) hasta chocar con un 1. A esta distancia promedio se la denomina longitud de correlación.

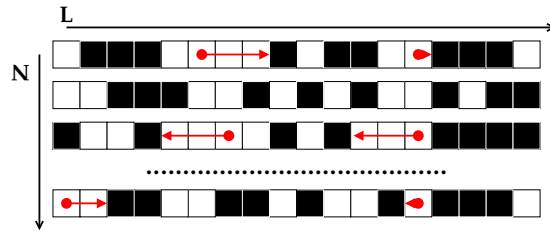


Figura 3: Matriz de poros y definición de longitud de correlación.

Ejercicio 3: Contar qué distancia podría recorrer un proceso de transporte que se iniciase en cada uno de los 0s de una cadena y se moviese hacia la derecha o hacia la izquierda. Genérense N cadenas de longitud L y porosidad r . Recorriendo cada cadena, cada vez que se encuentre un 0, calcúlense los números de 0s a su derecha e izquierda, calculando finalmente la media de todas esas distancias. Tómense $L = 1024$, $N = 1024$ y $r = 0,1, 0,25, 0,5, 1, 2, 4, 9$. Represéntese la

longitud de correlación en función de r . ¿Dependerá el resultado de L ? (pruébese con un $L = 32$). ¿Qué nos indica esto acerca de la elección de nuestro tamaño de simulación?

Ejercicio 4: Calcular las distancias medias de transporte por un método de Monte Carlo. Genérense, como antes, N cadenas de longitud L y porosidad r . Escójanse al azar $M = N/16$ de los 0s de cada una de esas cadenas y, recorriendo a derecha o izquierda (escogiendo el sentido también al azar) calcúlese la distancia hasta el primer 1. Hacer la media de todas estas $N \times M$ distancias calculadas y comparar el resultado con el obtenido por la búsqueda exhaustiva del ejercicio anterior.

Nota: En casos tan sencillos como el de la cadena unidimensional, es posible hacer un cálculo exacto de las distancias de correlación en un tiempo razonable. Pero cuando el tamaño de la cadena aumenta (p.ej. $L = 10^6$ sitios) o cuando tratamos con medios desordenados bi- o tridimensionales, el cálculo por Monte Carlo es la única alternativa factible.

Apéndice: “trucos” de C

1. En C existen diferentes tipos de datos numéricos. A diferencia de otros lenguajes de programación, el tipo de dato `char` o `unsigned char` es también numérico. Representa un número entero entre -127 y 128 (`char`) o entre 0 y 255 (`unsigned char`): lo suficiente para codificar cualquier carácter ANSI y codificación ampliada, como la Latin-1 o ISO-8859-1 que usamos habitualmente. Esta versatilidad puede servir a una función: el ahorro de memoria. Una matriz de 1024×1024 caracteres ocupa 1 Mb de memoria, que es el valor típico que se reserva para todas las variables locales en algunos entornos. Una matriz de ese tamaño de enteros (`int`) ocupa 16 Mb. Para guardar valores de 0 o 1 puede ser eficiente usar `chars`, para no estar limitados a matrices de tamaños muy pequeños (o tener que usar asignación dinámica de memoria).
2. El generador de números aleatorios parte siempre de un valor inicial o semilla. Si esta semilla es siempre la misma, un programa generará siempre la misma secuencia de números aleatorios. Para evitar esto, se suele partir de una semilla que cambie en cada realización del programa y la más simple de estas semillas es el reloj del sistema. Así, una semilla apropiada (que cambia cada segundo) puede calcularse `#incluyendo` la `<time.h>` y haciendo

```
int semilla=(int)time(NULL);
```

Este valor se debe asignar a la semilla del generador de números aleatorios que se esté usando antes de empezar a generar valores con él. Es también

adecuado ejecutar un bucle de “termalización” del generador de números aleatorios después de asignar la semilla para evitar que los primeros números aleatorios generados en cada ejecución sean los obtenidos de una secuencia de números correlativos (los sucesivos segundos).

3. El generador de números aleatorios habitual de C es la función `rand()` de la biblioteca estándar de C o `<stdlib.h>`. Esta función retorna cada vez un número aleatorio entre 0 y `RAND_MAX`. Para calcular un número aleatorio en coma flotante en el intervalo $[0, 1)$ se hará

```
double rnd=(double)rand()/((double)RAND_MAX+1);
```

Para asignar la semilla del generador de números aleatorios `rand()` se usa la función `srand()` llamándola como

```
srand( semilla );
```