

**Curso:**  
**Métodos de Monte Carlo**  
**Unidad 3, Sesión 7: Problemas de conteo**

Departamento de Investigación Operativa  
Instituto de Computación, Facultad de Ingeniería  
Universidad de la República, Montevideo, Uruguay

dictado semestre 1 - 2021

## Problemas de conteo

Mientras que el cálculo de volúmenes es un problema muy frecuente en el contexto de espacios continuos, el conteo de objetos surge naturalmente en espacios discretos.

Consideremos un conjunto finito  $\mathcal{X}$ , llamado *conjunto base*, y una familia  $\mathcal{F} = \{\mathcal{S}_1, \dots, \mathcal{S}_k\}$  de subconjuntos de  $\mathcal{X}$ , llamada el *conjunto de objetos*. Muchos problemas encontrados frecuentemente pueden expresarse mediante la evaluación de

- $|\cup_{j=1}^k \mathcal{S}_j|$  = cantidad de elementos de  $\mathcal{X}$  que pertenecen al menos a un subconjunto de  $\mathcal{F}$ ,
- $|\cap_{j=1}^k \mathcal{S}_j|$  = cantidad de elementos de  $\mathcal{X}$  que pertenecen a la intersección de los subconjuntos de  $\mathcal{F}$ .

Daremos a continuación algunos ejemplos de estos problemas

## Sentencias lógicas

Sean  $m$  variables lógicas  $\mathbf{x} = (x_1, \dots, x_m)$  que toman valores en el conjunto base  $\mathcal{X} = \{0, 1\}^m$  de tamaño  $r = 2^m$ . Supongamos que existen  $k$  cláusulas o ecuaciones que estas variables deben satisfacer, y sea  $\mathcal{S}_j$  el subconjunto de  $\mathcal{X}$  que satisface la  $j$ -ésima cláusula.

Entonces la intersección  $|\cap_{j=1}^k \mathcal{S}_j|$  da la cantidad de sentencias lógicas (la cantidad de asignaciones de valores a las variables) que satisfacen al mismo tiempo todas las  $k$  cláusulas.

Supongamos que llamamos *solución factible* a un  $\mathbf{x} \in \mathcal{X}$  que satisface todas las cláusulas, y supongamos que queremos encontrar una solución factible  $\mathbf{x}^*$  que maximiza una cierta función  $w(\mathbf{x})$ . Muchos métodos para encontrar un óptimo llevan un tiempo que depende directamente del número total de soluciones factibles; el poder estimar por Monte Carlo el valor de  $|\cap_{j=1}^k \mathcal{S}_j|$  da una información importante para ver si estos métodos son aplicables (y estimar el tiempo de cálculo necesario), o en cambio determinar que es necesario aplicar una heurística o cálculo aproximado y quedarse con una solución “buena”, aunque no óptima.

## Conjuntos de desigualdades

Sea  $\mathbf{x} = (x_1, \dots, x_m)$  con  $x_i \in \{0, 1\}$ ,  $i = 1, \dots, m$ ; sea  $\mathbf{A} = ((a_{jl}))$  una matriz de dimensión  $k \times m$ ; sea  $\mathbf{b}$  un vector  $k \times 1$ . El problema es encontrar el número de soluciones para el conjunto de  $k$  desigualdades simultáneas siguientes:

$$\mathbf{Ax} \leq \mathbf{b}.$$

En este caso,  $\mathbf{x}$  toma valores en el conjunto base  $\mathcal{X} = \{0, 1\}^m$ , y el conjunto de objetos  $\mathcal{F}$  tiene como miembros

$$\mathcal{S}_j = \{\mathbf{x} \in \mathcal{X} : a_{j1}x_1 + \dots + a_{jm}x_m \leq b_j\},$$

para  $j = 1, \dots, k$ , de manera que  $|\cap_{j=1}^k \mathcal{S}_j|$  es el tamaño del conjunto de soluciones 0-1 que satisfacen todas las restricciones, lo que corresponde al conjunto de soluciones factibles en un problema de programación lineal 0-1 (un caso especial de la programación lineal entera).

## Probabilidad combinatoria

Sea  $\mathcal{X}$  un conjunto base de eventos aleatorios, y sea el conjunto de objetos  $\mathcal{F} = \{\mathcal{S}_1, \dots, \mathcal{S}_k\}$  conformado por el conjunto de subconjuntos de  $\mathcal{X}$  para los cuales el evento  $B$  ocurre (es verdadero).

Un problema clásico en probabilidad combinatoria es la evaluación de la probabilidad de ocurrencia de  $B$ , dada por

$$\text{Prob}(B) = \text{Prob}\left(\bigcup_{j=1}^k \mathcal{S}_j\right) = \sum_{r=1}^k (-1)^{r+1} \sum_{1 \leq j_1 < j_2 < \dots < j_r \leq k} \text{Prob}\left(\bigcap_{l=1}^r \mathcal{S}_{j_l}\right).$$

Nuevamente aquí vemos como aparecen problemas con el mismo formato previo, en los que métodos de conteo pueden resultar útiles.

## Métodos exactos versus estimación

Como estamos hablando en todos los casos de conjuntos discretos finitos, la forma que parece obvia para calcular los tamaños de los conjuntos unión e intersección parece ser la simple enumeración de todos los objetos del conjunto base, u otra forma exacta que genere de alguna forma los objetos deseados.

Sin embargo, una cantidad importante de estos problemas son *NP*-completos si consideramos como entrada las dimensiones de los mismos (y no la lista de elementos del conjunto base, que en general se da implícitamente y no explícitamente). Por lo tanto, los métodos enumerativos directos toman tiempos prohibitivos cuando la dimensión del caso base crece, y no son aplicables.

Esto motiva el empleo de otros métodos, algunos basados en la estructura específica del problema y sus propiedades teóricas, otros, como el uso de Monte Carlo, que son aplicables en todos los casos, pero dan un resultado aproximado en lugar del valor exacto.

## Esquema de un método Monte Carlo

Supongamos que queremos estimar  $\zeta = |\cup_{j=1}^k \mathcal{S}_j|$ , donde  $\mathcal{S}_1, \dots, \mathcal{S}_k$  son subconjuntos de  $\mathcal{X} = \{a_1, \dots, a_r\}$ . Los elementos  $a_j$  pueden representar cualquier objeto, en los ejemplos anteriores en general son vectores  $m$ -dimensionales en un espacio finito.

El algoritmo MonteCarlo-Unión, cuyo pseudocódigo damos a continuación, describe el método más simple y directo para estimar  $\zeta$  a través de un muestreo aleatorio. Para implementar este algoritmo, es necesario un procedimiento para generar  $a$  aleatoriamente y con probabilidad uniforme dentro del conjunto  $\mathcal{X}$ , esta tarea puede ser sencilla (como en el caso en que  $\mathcal{X} = \{0, 1\}^m$ , en que la generación de  $a$  lleva tiempo lineal en  $m$ ) o más compleja en otros casos. También es necesario, dado un  $a$ , poder controlar si  $a$  pertenece a algún  $\mathcal{S}_j$ , lo que comunmente lleva tiempo lineal en  $km$ .

El método se basa en generar una cantidad  $n$  de elementos de  $\mathcal{X}$ , y ver que proporción pertenece a  $\cup_{j=1}^k \mathcal{S}_j$ , para de allí estimar la proporción de

elementos de  $\mathcal{X}$  que pertenecen a esa unión; dado que conocemos el tamaño de  $\mathcal{X}$ , usando esa proporción es posible estimar directamente  $\zeta$ .

### Procedimiento MonteCarlo-Conteo

Entradas: conjuntos  $\mathcal{S}_j$ ,  $j \leq k$ ;  $n$  tamaño de la muestra;  $r = |\mathcal{X}|$ ; nivel de confianza  $1 - \delta$

Parámetros de salida:  $\bar{\zeta}_n$ , estimador de  $\zeta$ ;  $V(\bar{\zeta}_n)$ , estimador de  $\text{Var}(\bar{\zeta}_n)$ ; intervalo de confianza

1.  $S = 0$ . /\* Inicialización \*/
2. For  $i = 1, \dots, n$  do
  - 2.1 Sortear un  $a \in \mathcal{X}$  de forma aleatoria con probabilidad  $1/r$
  - 2.2 For  $h = 1, \dots, k$ 
    - 2.2.1 If  $a \in \mathcal{S}_h$  then  $S = S + 1$ ; break for;
3.  $\bar{\zeta}_n = rS/n$
5.  $V(\bar{\zeta}_n) = \bar{\zeta}_n(r - \bar{\zeta}_n)/(n - 1)$
6. Calcular  $(rI_1(S, n, \delta), rI_2(S, n, \delta))$  un intervalo de confianza de nivel  $1 - \delta$ .



Existen  $\zeta = |\cup_{j=1}^k \mathcal{S}_j|$  elementos de interés dentro de  $\mathcal{X}$  (que recordemos tiene cardinalidad  $r$ ), por lo tanto la probabilidad que  $a$ , elegido de manera uniforme (con probabilidad  $1/r$ ) dentro de  $\mathcal{X}$  pertenezca a  $\cup_{j=1}^k \mathcal{S}_j$  es  $\zeta/r$ .

Por lo tanto, es fácil ver que la esperanza de  $\bar{\zeta}_n$  es efectivamente  $\zeta$ . Además,  $S$  tiene distribución binomial de parámetro  $\lambda = \zeta/r$ , y varianza  $n\lambda(1 - \lambda)$ . Como  $\bar{\zeta}_n = rS/n$ , entonces  $\text{Var}(\bar{\zeta}_n) = r^2/n^2 \text{Var}(S)$ , y por lo tanto  $\text{Var}(\bar{\zeta}_n) = \zeta(r - \zeta)/n \leq r^2/4n$ .

Dado que  $S$  tiene la misma distribución que en el caso de la estimación de volúmenes, toda la discusión realizada en las sesiones anteriores sobre fijación de tamaño de muestra y sobre determinación de intervalos de confianza es aplicable directamente a este caso.

## Ejemplo: confiabilidad de redes

Veremos ahora un ejemplo basado en el problema de calcular la probabilidad de que dos nodos puedan comunicarse en una red donde las aristas están sujetas a fallas aleatorias e independientes (este problema es usualmente conocido como problema del cálculo de la confiabilidad fuente-terminal de una red, y ha recibido mucha atención en la literatura especializada).

Definimos formalmente el problema:

- Consideramos conocido una red dada por sus nodos y sus líneas de comunicación, y representada por un grafo  $G = (V, E)$ , donde  $V$  el conjunto de nodos del grafo coincide con el conjunto de nodos de la red, y  $E$  el conjunto de aristas del grafo coincide con el conjunto de las líneas de comunicación (consideramos el caso no orientado, en el que la comunicación es bidireccional).
- Hay dos nodos especiales,  $s$  y  $t$ , llamados fuente y terminal.

- Una arista se puede representar por sus dos nodos extremo. Una arista puede estar en funcionamiento o fallada, definimos  $x_e$  la variable aleatoria “estado de la arista  $e$ ”. Esta variable aleatoria tiene una distribución de Bernoulli de parámetro  $r_e$  la confiabilidad de  $e$ , y por lo tanto puede tomar los dos valores 0 o 1, donde  $x_e = 1$  si la arista funciona, y  $x_e = 0$  si la arista está fallada. Suponemos que las distintas aristas son independientes, y que los nodos no fallan.
- Decimos que la red funciona cuando existe un camino entre  $s$  y  $t$  conformado por aristas en funcionamiento, y que la red falla o no funciona en caso contrario. Denotamos  $\phi_{st}(\mathbf{x})$  la función de estructura de la red, que vale 1 cuando ésta funciona y 0 sinó.
- La medida de confiabilidad fuente-terminal,  $R_{st}(G)$ , es la probabilidad de  $s$  y  $t$  estén conectados en este modelo. Formalmente, si  $m = |E|$ , tenemos un conjunto base  $\mathcal{X} = \{0, 1\}^m$  formado por todos los estados posibles del grafo (entendiendo como estado del grafo a un vector  $\mathbf{x} = (x_1, \dots, x_m)$  que incluye el estado de cada una de las aristas).

Podemos calcular por la expresión  $\text{Prob}(\mathbf{x}) = \prod_{i/x_i=1} r_i \prod_{i/x_i=0} (1 - r_i)$  la probabilidad de observar el estado  $\mathbf{x}$ , entonces tenemos que

$$R_{st}(G) = \sum_{\mathbf{x} \in \mathcal{X}} \phi_{st}(\mathbf{x}) \text{Prob}(\mathbf{x}).$$

Esta expresión, si bien muy simple, implica sumar  $2^m$  términos, y por lo tanto crece exponencialmente con el número de aristas,  $m$ . Por lo tanto, en la práctica sólo resulta útil para grafos con muy pocas aristas, en el orden de unas pocas decenas.

## Estimación de la confiabilidad empleando conteos de conjuntos

Vamos a suponer que todas las aristas tienen la misma probabilidad de funcionamiento  $r_e = p$ . Entonces, la probabilidad de observar un estado  $\mathbf{x}$  se puede calcular simplemente a partir de la cantidad de aristas en funcionamiento y falladas que lo integran, si  $j = |\{e/x_e = 1\}| = \sum_{e=1}^m x_e$  (y por lo tanto,  $m - j = |\{e/x_e = 0\}|$ ), entonces

$$\text{Prob}(\mathbf{x}) = p^j (1 - p)^{m-j}.$$

Definimos ahora  $\mathcal{S}_j = \{\mathbf{x} | \phi_{st}(\mathbf{x}) = 1 \text{ y } \sum_{e=1}^m x_e = j\}$ , para todo  $j = 0, \dots, m$ , y  $c_j = |\mathcal{S}_j|$  los cardinales de estos conjuntos. Para un  $j$  dado,  $\mathcal{S}_j$  es el conjunto de los estados con  $j$  aristas en funcionamiento tales que  $s$  y  $t$  pueden conectarse.

Podemos reescribir la fórmula para la confiabilidad,

$$R_{st}(G) = \sum_{\mathbf{x} \in \mathcal{X}} \phi_{st}(\mathbf{x}) \text{Prob}(\mathbf{x}),$$

de la manera siguiente:

$$R_{st}(G) = \sum_{j=0}^m c_j p^j (1-p)^{m-j}.$$

Hemos logrado pasar de una suma de  $2^m$  términos a una suma de  $m + 1$  términos; por lo tanto, si empleamos algún método (por ejemplo Monte Carlo) para calcular o estimar los  $m + 1$  valores  $c_0, \dots, c_m$ , podemos calcular luego la confiabilidad fuente-terminal de  $G$  para cualquier valor de  $p$  de manera eficiente.

Para hacer la estimación de los  $c_j$  por Monte Carlo, es necesario poder determinar rápidamente si un estado  $\mathbf{x}$  pertenece o no a  $\mathcal{S}_j$ . Esto se puede

hacer en tiempo lineal en  $m$ , alcanza por un lado sumar los estados de las aristas (para ver si la suma es o no igual a  $j$ ), y por otro controlar si  $s$  y  $t$  están conectados, lo que se logra por ejemplo mediante una búsqueda Depth First Search (DFS) en el subgrafo formado por las aristas que funcionan.

## Preguntas para auto-estudio

- ¿Qué es un problema de conteo? ¿Cuáles son algunos ejemplos típicos?
- ¿Porqué puede no ser aplicable un método enumerativo para resolver un problema de conteo?
- ¿Cómo es el esquema de un método Monte Carlo para resolver un problema de conteo?



## Entrega 4

### Ejercicio 7.1 (individual):

Problema: supongamos que un programa de posgrado incluye un conjunto de  $P$  profesores y otro de  $S$  estudiantes, y que se desea asignar a cada estudiante un profesor para consultas, pero que todas las personas son de diferentes países y comprenden distintos idiomas. Si cada estudiante tiene asignado un profesor para consultas, la cantidad total de formas de asignar profesores a los estudiantes es  $P^S$  (hay  $P$  opciones para cada estudiante, y  $S$  estudiantes en total, la cantidad total de opciones es el producto sobre los estudiantes de las opciones de cada estudiante).

Supongamos que nos interesa determinar cuantas formas hay de asignar profesores a los estudiantes, respetando que cada profesor asignado a un estudiante tenga un idioma en común con él (para que puedan comunicarse). Si hay  $L$  lenguajes posibles, podemos tener para cada estudiante  $s$  un subconjunto  $Id(s)$  con los lenguajes que entiende, y lo mismo para cada profesor  $p$ ; y una asignación sólo sería válida si para cada

estudiante  $s$  y su profesor  $p(s)$ , se cumple que  $Id(s) \cap Id(p(s)) \neq \emptyset$  (estos conjuntos también podrían representarse como matrices  $Id_{s,l}$  e  $Id_{p,l}$ , con entradas 1 cuando ese estudiante o profesor entienden el lenguaje  $l$  y 0 si no).

Para estimar la cantidad de formas distintas de realizar estas asignaciones de profesores y estudiantes, es posible aplicar el método Monte Carlo.

Se debe recibir en entrada el número de replicaciones a realizar, y el nivel de confianza; en salida, se debe dar la estimación del número de combinaciones  $N_C$ , así como la desviación estándar y un intervalo de confianza (del nivel especificado) calculado en base al criterio de Agresti-Coull.

- Parte a: escribir un programa para hacer el cálculo previamente descrito. Entregar pseudocódigo y código.
- Parte b: sea el siguiente caso: Estudiantes/lenguajes: Maria: español, inglés; Sophie: inglés, francés; Liliana: español, portugués; Lucia:

inglés, portugués; Monique: francés; Rodrigo: español, inglés, francés; John: inglés; Neymar: portugués, español; Jacques: francés, portugués; Juan: español. Profesores: Tom: inglés, francés, español; Luciana: inglés, portugués; Gerard: inglés, francés; Silvia: español, francés.

Usando el programa anterior, y empleando 1000 replicaciones de Monte Carlo, estimar los valores de cuantas combinaciones  $N_C$  hay para asignar profesores a estudiantes y que tengan un idioma en común, con intervalos de confianza de nivel 95%.

- Parte c: adaptar el programa para calcular el número de combinaciones si además queremos agregar como restricción que ningún profesor atienda menos de un estudiante ni más de cuatro estudiantes.

Usando el programa modificado, y empleando 1000 replicaciones de Monte Carlo, estimar los valores de cuantas combinaciones  $N_C$  hay para asignar profesores a estudiantes y que tengan un idioma en común y cuplan la restricción adicional mencionada, con intervalos de confianza de nivel 95%.

Fecha entrega: Ver cronograma y avance del curso.