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 - 2024

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 $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ 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 $\mathbf{x} = (x_1, \ldots, x_m)$ 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 $\mathbf{x} = (x_1, \ldots, x_m)$ con $x_i \in \{0, 1\}$, $i = 1, \ldots, m$; 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

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

para $j = 1, \ldots, k$, 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 $\mathcal{F} = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ 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

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

Nuevamente aquí vemos como aparecen problemas con el mismo formato previo, en los que métodos de 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 $\mathcal{S}_1, \ldots, \mathcal{S}_k$ son subconjuntos de $\mathcal{X} = \{a_1, \ldots, a_r\}$. 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 seudocódigo damos a continuación, describe el método más simple y directo para estimar ζ 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 ζ.

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 ζ; $V(\bar{\zeta}_n)$, estimador de Var $\left(\bar{\zeta}_n\right)$; intervalo de confianza

1. $S = 0$. /* Inicialización */
2. For $i = 1, \ldots, n$ do
    2.1 Sortear un $a \in \mathcal{X}$ de forma aleatoria con probabilidad $1/r$
    2.2 For $h = 1, \ldots, k$
        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 ζ/r.

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

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 $\mathbf{x} = (x_1, \ldots, x_m)$ 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 $j = 0, \ldots, m$, 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 $c_0, \ldots, c_m$, 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: para diseñar un Sistema Nacional de Áreas Protegidas, uno de los modelos empleados tiene en cuenta por un lado un conjunto $E = \{1, \ldots, e\}$ de especies que se desea preservar, y por otra parte un conjunto Z de zonas donde es posible establecer una reserva. La relación entre ambos conjuntos está dada por una matriz $P = ((p_{ij}))$, con $i \in Z$ and $j \in E$, tal que $z_{ij} = 1$ ssi en la zona i se ha observado la presencia de la especie j.

Es interesante elegir un conjunto de M zonas, tales que todas las especies estén representadas en al menos una zona. La determinación del menor M que hace posible esta propiedad es un problema de "set covering" (NP-difícil), que escapa el alcance de este curso.

Suponiendo que el valor de M ya fue elegido, un segundo nivel es ver cuántas formas distintas hay de elegir este conjunto de zonas.

Para esto, es posible aplicar el método Monte Carlo para, dado el cardinal de E, el cardinal de Z, la matriz $P = ((p_{ij}))$, y un valor de M predeterminado, estimar cuántas combinaciones de M zonas distintas cumplen la propiedad requerida (representar todas las especies).

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 $N_C(M)$, 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 seudocódigo y código.
  Comentario: para el muestreo uniforme de los subconjuntos de Z, se debe hacer un "muestreo sin remplazos´´de las distintas zonas. Es posible hacerlo sorteando siempre números uniformes entre 1 y z, y descartando aquellos ya elegidos, aunque se pierde eficiencia por los sorteos repetidos. Otra forma más eficiente es es elegir primero de forma uniforme un número de 1 a z, y eliminar esta zona de Z; el

segundo sorteo hacerlo entre 1 y $z-1$; el tercero entre 1 y $z-2$, y así sucesivamente, donde en cada etapa sólo se elige entre las zonas que todavía no fueron sorteadas.

- Parte b: sea el siguiente caso: $z = 15$ zonas, $e = 8$ especies, $p_{ij} = 1$ para

$$\{(1,1),(1,2),(1,3),(1,6),(1,8),$$
$$(2,1),(2,3),(2,4),(2,6),(2,7),$$
$$(3,1),(3,4),(3,5),(3,7),(3,8),$$
$$(4,1),(4,2),(4,4),(4,6),(4,7),(4,8),$$
$$(5,1),(5,3),(5,6),(5,8),$$
$$(6,2),(6,4),(6,7),(6,8),$$
$$(7,3),(7,5),(7,8)$$
$$(8,2),(8,3),(8,5),(8,7),(8,8),$$

$$(9,2),(9,5),(9,6),(9,8),$$
$$(10,3),(10,4),(10,6),(10,7),(10,8)$$
$$(11,2),(11,5),(11,6),(11,7),$$
$$(12,3),(12,4),(12,5),(12,6),(12,7),$$
$$(13,1),(13,2),(13,6),(13,7),(13,8),$$
$$(14,1),(14,2),(14,4),(14,6),(14,7),$$
$$(15,2),(15,3),(15,5),(15,6),(15,7),(15,8)\}.$$

Comentario: este es un ejemplo "de juguete", los casos reales cubren cientos de zonas y también cientos de especies (animales y vegetales), y paisajes, a proteger. En la revista Enlaces.fing, número 4 (mayo 2010), pag. 23 a la 27, pueden encontrar una descripción (no técnica) del caso uruguayo:
`http://www.ricaldoni.org.uy/images/enlaces/enlaces_04.pdf`
(último acceso - 10-4-2019).

Usando el programa anterior, y empleando 1000 replicaciones de Monte

Carlo, estimar los valores de $N_C(M)$ para $M = 5$ y para $M = 6$ con intervalos de confianza de nivel 95%.

Fecha entrega: Ver cronograma y avance del curso.