Caminos mínimos

Algoritmos y Estructuras de Datos III

Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires

27 de septiembre de 2017

Floyd-Warshall

Camino mínimo entre todos los vértices

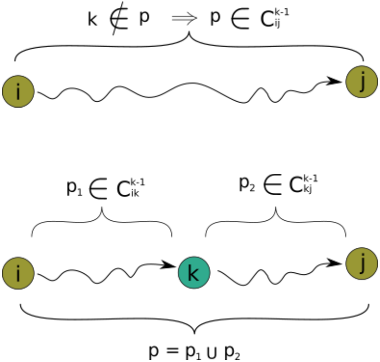
- Formulación de programación dinámica para resolver el problema.
- Fácil de implementar.

Floyd-Warshall

Sea G un grafo con vértices numerados $\{1, \ldots, n\}$

- Para cada par $i, j \in V$: Consideremos C_{ij}^{k} todos los caminos entre i, j qué pueden usar sólo los vértices $\{1, \ldots, k\}$, y sea p un camino mínimo de C_{ij}^{k}.
- ¿Qué relación hay entre C^{k-1} y p, donde $C^{k-1} = \{C_{lp}^{k-1} \forall l, p \in V\}$?

Estructura del camino mínimo



Formulación

Sea f(i,j,k) el peso del camino mínimo de *i* a *j* usando como vértices intermedios $\{1, 2, \ldots, k\}$

$$f(i,j,k) = \begin{cases} w_{ij} & \text{if } k = 0 \\ min\{f(i,j,k-1), f(i,k,k-1) + f(k,j,k-1)\} & \text{if } k > 0 \end{cases}$$

La solución del camino mínimo para un par de nodos *i*, *j* es f(i,j,n) donde *n* es la cantidad de nodos del grafo.

Algorithm 1: Floyd-Warshall

Data: $G = W$ (matriz de adyacencias)
Result: d (matriz de distancias)

$d_{ij}^0 \leftarrow W_{ij}$
$d_{ii}^0 \leftarrow 0$
for $k \leftarrow 1$ **to** n **do**
    for $i \leftarrow 1$ **to** n **do**
        for $j \leftarrow 1$ **to** n **do**
            if $d_{ij}^{k-1} > d_{ik}^{k-1} + d_{kj}^{k-1}$ **then**
                $d_{ij}^{k} \leftarrow d_{ik}^{k-1} + d_{kj}^{k-1}$
            else
                $d_{ij}^{k} \leftarrow d_{ij}^{k-1}$
        end
    end
end

Problemas

- ¿Complejidad temporal? $\mathcal{O}(n^3)$
- ¿Qué pasa si el grafo tiene aristas de peso negativo?
- ¿Sirve para detectar ciclos?
- ¿Complejidad espacial? $\mathcal{O}(n^3)$ ¿Se puede mejorar?

Detectando ciclos negativos

1. d_{ii} representa el camino mínimo para ir de i a i y es correctamente inicializado en 0.
2. La posición d_{ii} solo puede ser modificada cuando $j = i \wedge d_{ii} > d_{ik} + d_{ki}$.
3. Por 1 y 2 tenemos que la primera vez que se modifica $0 > d_{ik} + d_{ki}$, indicando que existe un vértice k para el cual la suma de los caminos mínimos $i \rightsquigarrow k$ y $k \rightsquigarrow i$ es negativo.
4. Cuando esto sucede por primera vez, el valor de d_{ii} pasa a ser negativo, y como siempre que se actualiza es porque existe un camino de menor peso, entonces se va a mantener negativo.
5. Por lo que para buscar ciclos negativos hay que mirar la diagonal de d para ver que no contenga valores negativos.
6. Si hay ciclos negativos el problema no está bien definido.

Complejidad espacial

En cada iteración se utiliza d_{ik}^{k-1} y d_{kj}^{k-1}

¿Cuándo se modifican estas posiciones en la iteración k?

- d_{ik}^{k} se modifica $\Leftrightarrow k = j \bigwedge d_{ij}^{k-1} > d_{ij}^{k-1} + d_{jj}^{k-1}$
- $\Leftrightarrow d_{jj}^{k-1} < 0 \Leftrightarrow$ hay ciclo de longitud negativa.
- Lo mismo vale para d_{kj}^{k}

Podemos entonces compartir la matriz de distancias de todas las iteraciones reduciendo la complejidad espacial a $\mathcal{O}(n^2)$.

Reconstrucción de los caminos

- Dado un camino mínimo $i \leadsto j$:
  ¿Qué cosas comparte con los caminos mínimos $i \leadsto k$ y $k \leadsto j$ si sabemos que $k \in i \leadsto j$?
- Comparten todos los nodos: $i \leadsto j = i \leadsto k \leadsto j$.
- En particular, el nodo siguiente a i en el camino $i \leadsto k$ es el mismo que el vértice siguiente a i en el camino $i \leadsto j$.
- También, el nodo siguiente a k en el camino $k \leadsto j$ es el mismo que el vértice siguiente a k en el camino $i \leadsto j$.

Reconstrucción de los caminos

Matriz next

$next_{ij}$ es el vértice siguiente a i en el camino $i \rightsquigarrow j$.

- **Inicialización**: $next_{ij} = j$ si existe el eje (i, j) en el grafo.
- **Actualización**: Cada vez que se decide que el camino mínimo para ir de i a j pasa por el vértice k, se guarda en $next_{ij}$ que el siguiente vértice para ir de i a j es el mismo que el ya calculado para ir de i a k.

Reconstrucción: El camino arranca con i, luego con el vértice q indicado en $next_{ij}$ ... ¿y después como sigue el camino? Los demás vértices son los mismos que los del camino de q a j, por lo que podemos usar $next_{qj}$ para saber cuál es el vértice siguiente, esto se repite hasta llegar a j.

Floyd-Warshall completo

Algorithm 2: Floyd-Warshall

Data: $G = W$ (matriz de adyacencias)
Result: d (matriz de distancias), *next* (matriz de caminos)

```
d_ij ← W_ij
d_ii ← 0
next_ij ← j
for k ← 1 to n do
    for i ← 1 to n do
        for j ← 1 to n do
            if d_ij > d_ik + d_kj then
                d_ij ← d_ik + d_kj
                next_ij ← next_ik
            end
        end
        if d_ii < 0 then
            Return 'Encontré ciclo de longitud negativa'
        end
    end
end
```

Dantzig

Camino mínimo entre todos los pares de vértices

- Misma idea que Floyd, pero calcula en otro orden
- Simplifica el recalculo de caminos mínimos al agregar un vértice al grafo

Clave

Al finalizar la iteración **k-1**, el algoritmo de Dantzig obtiene los caminos mínimos en el subgrafo inducido por los vértices $\{v_1, v_2, \ldots, v_k\}$

Dantzig

- Llegar de v_i hasta v_{k+1} es el mínimo de $v_i \rightsquigarrow v_j \rightsquigarrow v_{k+1}$ (para algún j)
- $L^{k+1}_{i(k+1)} = min_{1 \leq j \leq k}(L^k_{ij} + L^k_{j(k+1)})$
- Llegar de v_{k+1} hasta v_i es el mínimo de $v_{k+1} \rightsquigarrow v_j \rightsquigarrow v_i$ (para algún j)
- $L^{k+1}_{(k+1)i} = min_{1 \leq j \leq k}(L^k_{(k+1)j} + L^k_{ji})$
- Llegar de v_i hasta v_j es el mínimo entre: ($v_i \rightsquigarrow v_{k+1} \rightsquigarrow v_j$, $v_i \rightsquigarrow v_j$ (sin usar v_{k+1}))
- $L^{k+1}_{ij} = min(L^k_{ij}, L^k_{i(k+1)} + L^k_{(k+1)j})$
- Usamos el mismo sistema de Floyd-Warshall para reconstruir los caminos con la matriz next.

Algorithm 3: Dantzig

Data: $G = W$ (matriz de adyacencias)
Result: L

```
L_ij ← W_ij
for k ← 1 to n − 1 do
    for i ← 1 to k do
        L_i(k+1) ← min_{1≤j≤k}(L_ij + L_j(k+1))
        L_(k+1)i ← min_{1≤j≤k}(L_(k+1)j + L_ji)
    end
    t ← min_{1≤i≤k}(L_(k+1)i + L_i(k+1))
    if t < 0 then
        Return 'Encontré ciclo de longitud negativa'
    end
    for i ← 1 to k do
        for j ← 1 to k do
            L_ij ← min(L_ij, L_i(k+1) + L_(k+1)j)
        end
    end
end
```

Problema

Problema

Dado un grafo, calcular los caminos mínimos entre todos los pares de vértices usando Floyd.
Le agregamos un vértice al grafo, y queremos saber todas las distancias mínimas. ¿Cómo hacemos?

Dudas

¿ Preguntas ?