Floyd-Warshall Algorithm

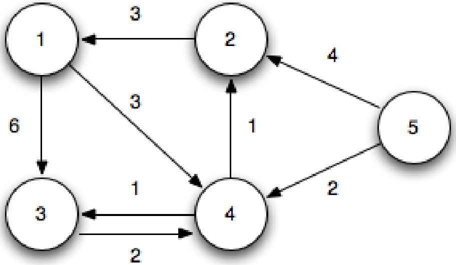
Chandler Burfield

February 20, 2013

All-Pairs Shortest Paths

Problem

To find the shortest path between all vertices $v \in V$ for a weighted graph $G = (V, E)$.



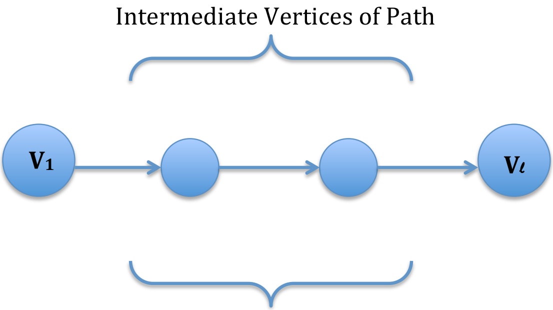
Dynamic-Programming

How to Develop a Dynamic-Programming Algorithm

1. Characterize the structure of an optimal solution.
2. Recursively define the value of an optimal solution.
3. Compute the value of an optimal solution in a bottom-up manner.

Intermediate Vertices

An **intermediate vertex** of a simple path $p = \{v_1, v_2, ..., v_l\}$ is any vertex other than v_1 or v_l i.e. a vertex from the set $\{v_2, ..., v_{l-1}\}$.



Weight Matrix

w_{ij}

The weight of an edge between vertex i and vertex j in graph $G = (V, E)$, where

$$w_{ij} = \begin{cases} 0 & \text{if } i = j \\ \text{the weight of a directed edge } (i, j) & \text{if } i \neq j \text{ and } (i, j) \in E \\ \infty & \text{if } i \neq j \text{ and } (i, j) \notin E \end{cases}$$

W

An $n \times n$ matrix representing the edge weights of an n-vertex graph, where $W = (w_{ij})$.

Path Matrix

$d_{ij}^{(k)}$

The weight of the shortest path from vertex i to vertex j for which all intermediate vertices are in the set $\{1, 2, ..., k\}$.

$D^{(k)}$

An n x n matrix representing the path distances between vertices in a directed n-vertex graph, where $D^{(k)} = (d_{ij}^{(k)})$.

Observations

- A shortest path does not contain the same vertex more than once.
- For a shortest path from i to j such that any intermediate vertices on the path are chosen from the set $\{1, 2, ..., k\}$, there are two possibilities:
  - 1. k is not a vertex on the path,
    so the shortest such path has length d_{ij}^{k-1}
  - 2. k is a vertex on the path,
    so the shortest such path is $d_{ik}^{k-1} + d_{kj}^{k-1}$
- So we see that we can recursively define $d_{ij}^{(k)}$ as

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k = 0 \\ min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & \text{if } k \geq 1 \end{cases}$$

Predecessor Matrix

$\pi_{ij}^{(k)}$

The predecessor of vertex j on a shortest path from vertex i with all intermediate vertices in the set $\{1, 2, ..., k\}$. Where

$$\pi_{ij}^{(0)} = \begin{cases} NIL & \text{if } i = j \text{ or } w_{ij} = \infty \\ i & \text{if } i \neq j \text{ and } w_{ij} < \infty \end{cases}$$

And

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \pi_{kj}^{(k-1)} & \text{if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \end{cases}$$

$\Pi^{(k)}$

The predecessor matrix, where $\Pi^{(k)} = (\pi_{ij}^{(k)})$.

Algorithm

Floyd-Warshall(W)

$n = W.rows$

$D^{(0)} = W$

for $k = 1$ to n

let $D^{(k)} = (d_{ij}^{(k)})$ be a new matrix

for $i = 1$ to n

for $j = 1$ to n

$d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$

return $D^{(n)}$

Analysis of Algorithm

Floyd-Warshall(W)

$n = W.rows$

$D^{(0)} = W$

for $k = 1$ to n

let $D^{(k)} = (d_{ij}^{(k)})$ be a new matrix

for $i = 1$ to n

for $j = 1$ to n

$d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$

return $D^{(n)}$

Runtime

$\Theta(n^3)$

Space

$\Theta(n^3)$ here,
but if we reuse space this can be done in $\Theta(n^2)$.

Analysis of Improved Algorithm

```
Floyd-Warshall(W)
    n = W.rows
    D = W
    Π initialization
    for k = 1 to n
        for i = 1 to n
            for j = 1 to n
                if d_ij > d_ik + d_kj
                then d_ij = d_ik + d_kj
                     π_ij = π_kj
    return D
```

Analysis

- The shortest path can be constructed, not just the lengths of the paths.
- Runtime: $\Theta(n^3)$.
- Space: $\Theta(n^2)$.

Floyd-Warshall(W)

$n = W.rows$

$D = W$

Π initialization

for $k = 1$ to n

for $i = 1$ to n

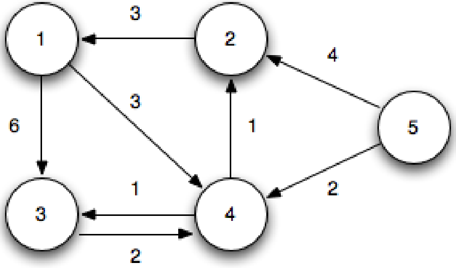
for $j = 1$ to n

if $d_{ij} > d_{ik} + d_{kj}$

then $d_{ij} = d_{ik} + d_{kj}$

$\pi_{ij} = \pi_{kj}$

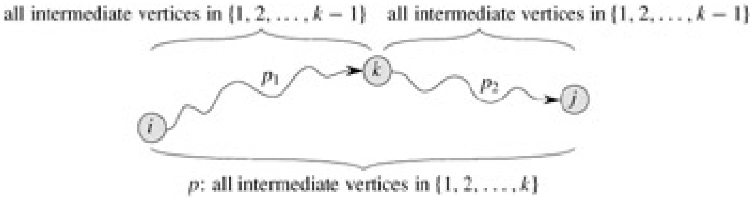
return D



Proof of Correctness

Inductive Hypothesis

Suppose that prior to the kth iteration it holds that for $i, j \in V$, d_{ij} contains the length of the shortest path Q from i to j in G containing only vertices in the set $\{1, 2, ..., k-1\}$, and π_{ij} contains the immediate predecessor of j on path Q.



Applications

- Detecting the Presence of a Negative Cycle
- Transitive Closure of a Directed Graph

Other All-Pairs Shortest Paths Algorithms

- Dynamic Programming Approach Based on Matrix Multiplication
- Johnson's Algorithm for Sparse Graphs