6.006 Lecture 16 Nov. 8, 2016

Today: All-Pairs Shortest Paths

- Floyd Warshall Algorithm
- Johnson's Algorithm
- difference constraints

Recall: Single-source Shortest Paths [last 3 lectures]

- given directed graph $G=(V,E)$
  node $s \in V$
  edge weights $w: E \to \mathbb{R}$
- find $\delta(s,v) \equiv$ shortest-path weight $s \to v$ $\forall v \in V$
  (or $-\infty$ if neg-weight cycle along way
  or ∞ if no path)

situation	best known algorithm	time
unweighted ($w=1$)	BFS	$O(V+E)$
nonneg edge weights	Dijkstra	$O(E+V \lg V)$ uses Fibonacci heaps
general	Bellman-Ford	$O(VE)$
acyclic graph (DAG)	topological sort + 1 pass B-F	$O(V+E)$

Today: All-pairs Shortest Paths

- given G, w
- find $\delta(u,v)$ for all $u,v \in V$

situation	algorithm	time	$E=\Theta(V^2)$
unweighted	$\|V\| \times$ BFS	$O(VE)$	$O(V^3)$
non neg weights	$\|V\| \times$ Dijkstra	$O(VE+V^2\lg V)$	$O(V^3)$
general	$\|V\| \times$ B-F	$O(V^2E)$	$O(V^4)$
* general	Floyd-Warshall	$O(V^3)$	$O(V^3)$
* general	Johnson's	$O(VE+V^2\lg V)$	$O(V^3)$

except third, all are best known!

$\boxed{\text{Cool thing: sometimes can actually do several computations at the same time faster than doing each separately!}}$

Application: Google Maps preprocessing (between waypoints)

Internet routing

- define $w(u,v)=\infty$ for $(u,v)\notin E$
  $w(u,u)=0$

Floyd Warshall Algorithm

- simple + efficient in practice !!
- Assumes no negative weight cycles

(but F-W can be used to detect neg wt cycles)

$C = (\omega(u,v))$ ⟸ matrix of weights

For $k = 1, 2, \ldots n$

  For u in V:

    For v in V:

Time: $O(V^3)$

      $\left.\begin{array}{l} \text{if } c_{uv} > c_{uk} + c_{kv} \\ c_{uv} = c_{uk} + c_{kv} \end{array}\right\}$ relaxation

Ok to omit superscripts telling us the iteration # k because more relaxation never hurts, but equivalently, could write as

$$c_{uv}^{(k)} = \min\{c_{uv}^{(k-1)},\ c_{uk}^{(k-1)} + c_{kv}^{(k-1)}\}$$

Correctness lemma:

After round k, $c_{uv}^{(k)} \le$ weight of shortest u-v-path using only intermediate nodes in $\{1 \ldots k\}$

Proof induction on k.

$k=0$: $c_{uv}^{(0)} = w(u,v)$ ✓ (direct edge is only path)

if true for round $\le k-1$:

  for u, v:

    let P be any min wt path from u to v using nodes in $\{1..k\}$

Two cases:

$k \notin p$ – then p is also a shortest path from u to v using nodes in $\{1..k-1\}$ } so $c_{uv}^{(k)} = c_{uv}^{(k-1)}$

$k \in p$ –

$$\overbrace{\underbrace{u \rightsquigarrow k}_{p_1} \; \underbrace{\rightsquigarrow v}_{p_2}}^{p}$$

(p_1 + p_2 do not contain k) ⇐ why? no negative weight cycles

as in Lecture 13 "optimal substructure" subpaths of shortest paths are shortest paths

p_1 is a shortest path from u to k with intermediate nodes in $\{1..k-1\}$

p_2 is a shortest path from k to v with intermediate nodes in $\{1..k-1\}$

So $c_{uv}^{(k)} = c_{uk}^{(k-1)} + c_{kv}^{(k-1)}$

So $c_{uv}^{(k)} = \min \{ c_{uv}^{(k-1)}, c_{uk}^{(k-1)} + c_{kv}^{(k-1)} \}$ $\square$

Main idea:
Change graph so that all new weights are positive, but shortest paths are the same.

Johnson's algorithm

1. find fnctn $h: V \to \mathbb{R}$ such that

$$w_h(u,v) \equiv w(u,v) + h(u) - h(v) \geq 0 \quad \text{for all } u,v \in V$$

<u>or</u> determine that negative-weight cycle exists.

2. Run Dijktra's algorithm on $G=(V,E)$ <u>with weights w_h</u>

from every source vertex $s \in V$

$\Rightarrow$ get $\delta_h(u,v)$ for all $u,v \in V$

3. Claim $\delta(u,v) = \delta_h(u,v) + h(u) - h(v)$

<u>Why is claim true?</u>

<u>Proof</u>:
- look at any $u \to v$ path p in G

p is $\underset{\overset{\shortparallel}{u}}{v_0} \to v_1 \to \ldots \to \underset{\overset{\shortparallel}{v}}{v_k}$

- what is $w_h(p)$?

$$w_h(p) = \sum_{i=1}^{k} w_h(v_{i-1}, v_i)$$

$$= \sum_{i=1}^{k} w(v_{i-1}, v_i) + h(v_{i-1}) - h(v_i)$$

$$= h(v_0) - \cancel{h(v_1)} + \cancel{h(v_1)} - \cancel{h(v_2)} + \cancel{h(v_2)} - h(v_3) + \ldots + \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

$$= \underbrace{h(v_0)}_{= h(u)} - \underbrace{h(v_k)}_{= h(v)} + \underbrace{\sum_{i=1}^{k} w(v_{i-1}, v_i)}_{w(p)}$$

$$= w(p) + h(u) - h(v)$$

$\Rightarrow$ so all u,v paths
change by same offset $= +h(u) - h(v)$

$\Rightarrow$ shortest (u,v)-path is preserved
(but length is offset) ∎

How to find h?

$\forall u,v \in V$:

$$w_h(u,v) \equiv w(u,v) + h(u) - h(v) \geq 0$$

← what we want

$$\Updownarrow$$

$$h(v) - h(u) \leq w(u,v)$$

← equivalent formulation

system of difference constraints

<u>Theorem</u>: if (V, E, w) has negative weight cycle then no solution to difference constraints

<u>Proof</u>: Say $v_0 \to v_1 \to \ldots \to v_k \to v_0$ is neg. weight

Suppose (for contradiction) that such a solution h exists:

$$
\begin{aligned}
h(v_1) - h(v_0) &\le w(v_0, v_1) \\
h(v_2) - h(v_1) &\le w(v_1, v_2) \\
&\vdots \\
h(v_k) - h(v_{k-1}) &\le w(v_{k-1}, v_k) \\
h(v_0) - h(v_k) &\le w(v_k, v_0)
\end{aligned}
$$

cancel

cancel

take sum of all rows

$$0 \le w(\text{cycle}) < 0$$

from assumption

So $0 < 0$? CONTRADICTION! $\square$

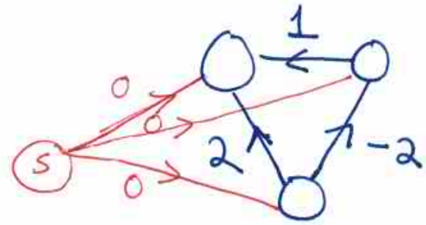
Theorem if (V,E,w) has no negative-weight cycle then can solve difference constraints

Proof

Algorithm to find h:

- Add to G a new vertex s
- add weight $= 0$ edges (s,v) for all $v \in V$
- Assign $h(v) = \delta(s,v)$ by doing Bellman-Ford from s

example



Note:
- no new cycles
  $\Rightarrow$ still no neg. weight cycles
- $s \Rightarrow v$ path exists $\Rightarrow \delta(s,v)$ finite $\forall\, v \in V$

Assign $h(v) = \delta(s,v)$:

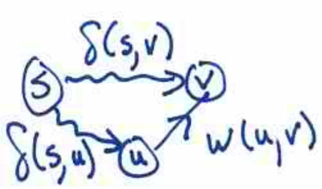
then $h(v) - h(u) \le w(u,v)$ ⇐ so this is true too!

$$\Updownarrow$$

$$\delta(s,v) - \delta(s,u) \le w(u,v)$$

$$\Updownarrow$$

$$\delta(s,v) \le \delta(s,u) + w(u,v)$$ ⇐ true, by $\triangle \neq$



$\square$

Analysis

①	Bellman-Ford from s	$O(VE)$
	+ reweight edges	$O(E)$
②	$\lvert V\rvert \times$ Dijkstra	$O(VE + V^2 \lg V)$
③	reweight all pairs	$O(V^2)$
		$O(VE + V^2 \lg V)$

Also:

Bellman-Ford can solve any system of difference constraints $\{x - y \le c\}$ or report "unsolvable" in $O(\underset{\uparrow}{V}\,\underset{\nwarrow}{E})$ ← # constraints

variables

Applications to real-time programming
multimedia scheduling
temporal reasoning

e.g. $LB \le t_{end} - t_{start} \le UB$