Weighted graphs

A weighted random number generator just produced a new batch of numbers.

Let's use them to build narratives!

All sports commentary
BFS (unweighted graphs)

Create FIFO queue to explore unvisited nodes
Dijkstra's algorithm is the BFS equivalent for non-negative weight graphs
Dijkstra

Dijkstra(G, w, s)
initialize G from s
Q = G.V, S = empty
while Q not empty
    u = Extract-min(Q)
    S = S U {u}
    for each v int G.Adj[u]
        relax(u, v, w)

S optional
Dijkstra

(a) Network Model

(b) Distance initialized

(c) Distance to adjacent nodes updated

(d) Node 3 selected

(e) Node 2 selected

(f) Node 5 selected

(g) Node 4 selected

(h) Shortest path found
Dijkstra

Runtime?
Dijkstra

Runtime:
Extract-min() run $|V|$ times
Relax runs Decrease-key() $|E|$ times
Both take $O(\lg n)$ time

So $O\left( |V| + |E| \right) \lg |V|$ time
(can get to $O(|V|\lg|V| + E)$ using Fibonacci heaps)
Runtime note:
If $G$ is almost fully connected, $|E| \approx |V|^2$

Use a simple array to store v.d
Extract-min() = $O(|V|)$
Decrease-key() = $O(1)$
total: $O(|V|^2 + E)$
Relaxation properties:
1. $\delta(s,v) \leq \delta(s,u) + \delta(u,v)$ (triangle inequality)
2. $v.d \geq \delta(s,v)$, $v.d$ is monotonically decreasing
3. if no path, $v.d = \delta(s,v) = \infty$
4. if $\delta(s,v)$, when $(v.\pi).d = \delta(s,v.\pi)$ then relax$(v.\pi,v,w)$ causes $v.d = \delta(s,v)$
5. if $\delta(v_0,v_k) = p_{0,k}$, then when relaxed in order $(v_0, v_1), (v_1, v_2), \ldots (v_{k-1}, v_k)$ then $v_k.d = \delta(v_0,v_k)$ even if other relax happen
6. when $v.d = \delta(s,v)$ for all $v$ in $V$, $G_{\pi}$ is shortest path tree rooted at $s$
Dijkstra

Correctness: (p.660)
Sufficient to prove when u added to S, u.d = δ(s,u)

Base: s added to S first, s.d=0=δ(s,s)

Termination: Loop ends after Q is empty, so V=S and we done
Dijkstra

Step: Assume v in S has v.d = δ(s,v)
Let y be the first vertex outside S on path of δ(s,u)

We know by relaxation property 4, that δ(s,y)=y.d (optimal sub-structure)

y.d = δ(s,y) ≤ δ(s,u) ≤ u.d, as w(p)≥0
Dijkstra

Step: Assume \( v \) in \( S \) has \( v.d = \delta(s,v) \). But as \( u \) was picked before \( y \), \( u.d \leq y.d \), combined with \( y.d \leq u.d \).

Thus \( y.d = \delta(s,y) = \delta(s,u) = u.d \).
Cycles

Does a shortest path need to contain a cycle?
Cycles

Does a shortest path need to contain a cycle?

No, case by cycle weight:
positive: why take the cycle?!
zero: can delete cycle and find same length path
negative: cannot ever leave cycle
Bellman-Ford

One of the few “brute force” algorithms that got a name

Idea:
1. Relax every edge (yes, all)
2. Repeat 1. $|V|$ times
Bellman-Ford

\[ BF(G, w, s) \]

initialize graph

for \( i = 1 \) to \(|V| - 1\) 

for each edge \((u, v)\) in \(G.E\) 

\[ \text{relax}(u, v, w) \]

for each edge \((u, v)\) in \(G.E\) 

if \( v.d > u.d + w(u, v) \): return false

return true
Bellman-Ford
Bellman-Ford

Correctness: (you prove)

After BF finishes: if \( \delta(s, u) \) exists, then \( \delta(s, u) = u.d \)
Bellman-Ford

Correctness: (you prove)

After BF finishes: if $\delta(s,u)$ exists, then $\delta(s,u) = u.d$

Relaxation property 5, as every edge is relaxed $|V|-1$ times and there are no loops
Bellman-Ford

Correctness: returns false if neg cycle
Suppose neg cycle: \( c = \langle v_0, v_1, \ldots, v_k \rangle \)
then \( w(c) < 0 \), suppose BF return true
Then \( v_i.d \leq v_{i-1}.d + w(v_{i-1}, v_i) \)
sum around cycle \( c \):
\[
\sum_{i=1}^{k} v_i.d \leq \sum_{i=1}^{k} (v_{i-1}.d + w(v_{i-1}, v_i))
\]
\[
\sum_{i=1}^{k} v_i.d \leq \sum_{i=1}^{k} v_{i-1}.d \text{ as loop}
\]
Bellman-Ford

Correctness: returns false if neg cycle

\[ \sum_{i=1}^{k} v_i.d \leq \sum_{i=1}^{k} (v_{i-1}.d + w(v_{i-1}, v_i)) \]

\[ \sum_{i=1}^{k} v_i.d = \sum_{i=1}^{k} v_{i-1}.d \text{ as loop} \]

so \( 0 \leq \sum_{i=1}^{k} w(v_{i-1}, v_i) \)

but \( \sum_{i=1}^{k} w(v_{i-1}, v_i) = w(c) < 0 \)

Contradiction!