Paths

September-10-13 8:29 AM

Terminology

Nodes and **Arcs**

Network = Directed Graph

Can be extended with labelled edges

Directed uv-path (definition 1)

A directed uv-path is a sequence $(u_0, u_1, \ldots, u_k)$ of vertices so that:
$u_0 = u,\ u_k = k$, and for $i = 1,2,\ldots,k$ there is an arc $u_{i-1}u_i$ in the directed graph; $u_0, u_1, \ldots, u_k$ are distinct.

Directed uv-path (definition 2)

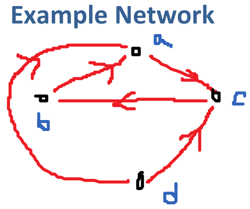
A sequence $a_1, a_2, \ldots, a_k$ of arcs such that for $i = 2,3,\ldots,k$ the "head" $h(a_{i-1})$ is the "tail" $t(a_i)$, $h(a_1) = u, h(a_k) = v$
All the "ends" of the a_i are distinct.

Note

There can be multiple arcs between the same nodes.
Both are distinct paths:

There are 2 ab-paths in the above example

Example Network



Shortest Paths

Every arc a has a "length" l_a
Problem: Given nodes u and v, find the shortest total length directed uv-path.

Example Paths

In the example above
(d,c,a)
(d,c,b,a)

Dijkstra's Algorithm

September-11-13 12:29 PM

Algorithm

We are trying to find a shortest direct uv-path.
Each arc of the network has a length $w_a \geq 0$

1) Initialize label u with $(0,*)$ and every other node with $(\infty,*)$
2) While there is an arc xy so that the labels on x and y are (l_x, s) and (l_y, t) so that $l_y > l_x + w_{xy}$, relabel y with $(l_x + w_{xy}, x)$

Claim

If node x has label (l_x, s) at any point of the algorithm, then either $l_x = \infty$ or there is a directed ux-path of length l_x whose last arc is sx.

Note 1

When the algorithm terminates, then, for every arc xy, $l_y \leq l_x + w_{xy}$

After each iteration, we construct a subnetwork consisting of those nodes x for which $l_x < \infty$ and all arcs sx for which the label of x has second coordinate s.
[x has label (l_x, s) at the end of the iteration. If $l_x < \infty$, then include x and the arc sx...]

Lemma 1

The subnetwork described in note 1 is a tree, outdirected from u.

Proof of Lemma 1

Induction Step

After iteration i, we have a tree T_i outdirected from u.
We need to prove after iteration $i+1$, we have a tree outdirected from u.

How does T_i change to get us the new network?
Case 1: y is not in T_i
    In this case, T_{i+1} is T_i plus y and the arc xy.
Case 2: y is in T_i
    In T_i there is an arc zy and $T_{i+1} = (T_i - zy) + xy$
    We replace arc zy with arc xy.

We see that the new network has the right number of arcs compared to nodes to be a tree (because these numbers are the same as in the preceding network, which is assumed to be a tree). So it is enough to show, for every node v, there is an outdirected uv-path.

There was a directed uv-path in the preceding network; if zy is not in that path, then there is a directed uv-path (the same one) in the new network.
So suppose zy is in the directed uv-path P in the preceding network.
Let Q be a directed ux-path in the preceding network. We will show that Q is also in the new network and does not contain any node in the yv-subpath of P.

Then Q from u to x, xy and P from y to v is the desired uv-path in the new network.

If zy is in Q then y is a node of Q. Se show more generally that no node of the yv-subpath of P is in Q (this includes showing y is not in Q).
Let w be a node of the yv-subpath of P that is in Q.
$l_x = \text{length}(Q) = \text{length}(Q[u,w]) + \text{length}(Q[w,x]) \geq \text{length}(Q[u,w]) = \text{length}(P[u,w])$
$\geq \text{length}(P[u,y]) = l_y$
But $l_y > l_x + w_{xy} \geq l_x$, so we have a contradiction.

We were trying to show that, after each iteration, the network consisting of the nodes with finite length label and arcs sx (for all x, x having label (l_x, s), with $l_x < \infty$) , is a tree outdirected from u.
Outdirected means: for each node v there is a directed uv-path.

Application: Production Line

- Ordered sequence of tasks to do to process out units.
- The i^{th} state has a probability α_i of introducing a defect.
- We can choose to inspect all or none of the units at various states in the production.
- At the end, we inspect everything.

Early detection saves manufacturing costs at the expense of inspection costs.

The relevant issues:

- There is a fixed cost f_{ij} for setting up an inspection station after stage j assuming the pervious inspection was at stage i.
- There is a per-unit cost g_{ij} of inspection after stage j assuming previous inspection at stage i $(i < j)$
- There is a per-unit manufacturing cost of p_i for stage i

We have network with $n+1$ nodes. An arc from node i to j represents doing an inspection at step j after having inspected at i.
Assuming we started the production line on B items, the cost of arc ij is

$$f_{ij} + B\left(\prod_{k=1}^{i}(1-\alpha_k)\right)\left(g_{ij} + \sum_{l=i+1}^{j} p_l\right)$$

Want to find min-cost path from 0 to n.

Min-Cost W. Neg. Weights

September-16-13 3:50 PM

Lemma 1

Let D be a directed graph with each arc a having a length w_a (possibly negative). Then either there are node labels y_v so that for each arc w, $y_v \leq y_u + w_{uv}$ for D has a negative cycle, but not both.

Lemma 2

Let P be a directed sx-path with length y_x. Suppose z is a node on P and the length of the sz-subpath of P is at least y_z and the length of the arc xz is w_{xz}.
If $y_z > y_x + w_{xz}$, then there is a negative cycle.

Min-cost with negative weights

In the case of all arc lengths ≥ 0, we have the following observation:
If l_v is the length of the shortest directed uv-path the, for every arc xy, $l_y \leq l_x + w_{xy}$

Suppose we have a label l_v on each node v of our network so that $l_u = 0$ and, for every arc xy, $l_y \leq l_x + w_{xy}$.
Let P be a directed uv-path $P = (v_0 = u, v_1, v_2, \dots, v_r = v)$
$l_{v_1} \leq l_{v_0} + w_{v_0 v_1}, l_{v_2} \leq l_{v_1} + w_{v_1 v_2}, etc.$
Adding them up, we find
$l_{v_1} + l_{v_2} + \dots + l_{v_k} \leq (l_{v_0} + w_{v_0 v_1}) + \dots + (l_{v_{k-1}} + w_{v_{k-1} v_k})$
$l_{v_k} \leq l_{v_0} + w_{v_0 v_1} + \dots + w_{v_{k-1} v_k} = 0 + \text{length of } P$
This depends on P being a directed uv-walk, not necessarily a path.

Example Negative Cycle

Currency trades:
$1 \xrightarrow{r_{1,2}} 2 \xrightarrow{r_{2,3}} 3 \xrightarrow{r_{3,4}} 4 \xrightarrow{r_{4,5}} 5 \xrightarrow{r_{5,1}} 1$
If we make these trades, we hope $r_{5,1} r_{4,5} r_{3,4} r_{2,3} r_{1,2} > 1$
Alternately, we want the sum of the $\log(r_i, r_{i+1})$ to be > 0
So setting the length of ij to $-\log(r, j)$ we are looking for a negative cycle.

Proof of Lemma 1

Not both is proved by assuming both exist and obtaining a contradiction. Let $(v_0, v_1, .., v_k - 1, v_0)$ be a negative cycle and suppose there are node weights so that, for each arc uv, $y_v \leq y_u + w_{uv}$.
In particular, for $i = 1, \dots, k$ (taking v_k to be v_o)

$$y_{v_i} \leq y_{v_{i-1}} + w_{v_{i-1} v_i}$$

Add these up to get

$$\sum_{i=1}^{k} y_{v_i} \leq \sum_{i=1}^{k} (y_{v_{i-1}} + w_{v_{i-1} v_i}) = \sum_{i=1}^{k} y_{v_{i-1}} + \sum_{i=1}^{k} w_{v_{i-1} v_i}$$

Since $v_0 = v_k$

$$\sum_{i=1}^{k} y_{v_i} = \sum_{i=1}^{k} y_{v_{i-1}} \text{ and , therefore}$$

$$\sum_{i=1}^{,} w_{v_{i-1} v_i} \geq 0$$

But $\sum_{i=1}^{k} w_{v_{i-1} v_i}$ is the length of the cycle, contradicting that it is a negative cycle.

To show either the node labels exist or there is a negative cycle, we consider very particular labels: fix u and y_v be the length of a shortest directed uv-path.

One issue: What if there is no directed uv-path? In this case v gets no label.

Let X be the set of nodes v for which there is a directed uv-path. Then every arc with exactly one end in X has its head in X.

Claim

If there are feasible node weights for $D[x]$ and for $D - X$, there are feasible node weights for D.

Proof

Let $u_1 v_1, u_2 v_2, \dots, u_r v_r$ be the arcs with one end in X and one end not in X. Then each $u_i \notin X$ and each $v_i \in X$. Let $d_i = y_{v_i} - (y_{u_i} + w_{u_i v_i})$. If every $d_i \leq 0$, then the node weights are already feasible for D. Otherwise, let $d = \max\{d_1, \dots, d_r\}$ and replace, for each $x \in X$ y_x with $y_x - d$

The digression shows we can assume every node can be reached from u. We show either these shortest path labels satisfy all inequalities $y_z \leq y_x + w_{xz}$ or there is a negative cycle.

Suppose there is an arc xz so that $y_z > y_x + w_{xz}$
In this case, we should find a negative cycle. Proved by lemma 2.

Proof of Lemma 2

Let C be the directed cycle consisting of the zx-subpath of P and the arc xz. Then $\text{lengt(C)} = \text{length}(P[z,x]) + \text{length}(xz) = \text{length}(P) - \text{length}(P[s,z]) + w_{xz} \leq y_x - y_z + w_{xz} < 0$. ∎

Bellman-Ford

September-20-13 12:48 PM

Theorem

At the termination of Bellman-Ford:

i) If the final node weights y_v are feasible, then y_v is the length of a shortest directed sv-path; and

ii) if the final node weights are infeasible, then; for any arc xz for which $y_z > y_x + w_{xz}$, there is a negative cycle containing xz.

We will find either shortest paths or a negative cycle.
In Bellman-Ford, we will find shortest directed walks from s to every other node among all directed walks having at most $|N| - 1$ arcs.
At the end of the algorithm, we will have either feasible node weights or a negative cycle. If we get feasible node weights, then the weights will be the lengths of the shortest directed paths from s.

We start at s and do a breadth first search going outwards.

Bellman-Ford

Initialize: $y_S = 0$; for every other node $v, y_v = +\infty$
For $r = 1,2, \ldots, |N| - 1$
    For each arc xz
        if $y_z > y_x + w_{xz}$ replace y_z with $y_x + w_{xz}$

We will prove: at the end, if node weights are feasible, then they are length so shortest paths; if not feasible, we will explain how to get a negative cycle.

Main Observation

After the r^{th}-iteration, the node weights are lengths of directed walks from s, and is at most the length of any such walk with r arcs.

Proof by induction on r: When a node label is changed from y_z to $y_x + w_{xz}$ the induction gives a directed sx-walk W of length y_x. Appending xz to W gives a directed sz-walk of length $y_x + w_{xz}$.
For the second part, if $W = (a_1, a_2, \ldots, a_t)$ is a directed sx-walk with $t \le r$ arcs, then we need to show length$(W) \ge y_x$ at the end of iteration r.

Note that the node weights can only decrease; they never increase.
    If $t < r$, then induction shows that after iteration $r - 1$, $r_x^{(r-1)} \le$ length(P)
        Since $y_x^{(r)} \le y_x^{(r-1)}$, we see that $y_x^{(r)} \le$ length(P)
    If $t = r$, then set $W' = (a_1, a_2, \ldots, a_{t-1})$. This is a directed sz-walk. After iteration $r - 1$, length$(W') \ge y_z^{(r-1)}$. When we consider zx in iteration r, we will arrange $y_x^{(r_{mid})} \le y_z^{(r_{mid})} + w_{zx} \le$ length(W)

Proof of Theorem

To prove (i), we suppose the node weights are feasible. By an earlier observation, this implies that there is no negative cycle.
For each node v, y_v is at most the length of any directed sv-walk having at most $|N| - 1$ arcs. (Lemma from last lecture.) Also, there is a directed sv-walk W_v of length y_v.

Every directed sv-path is a directed sv-walk with at most $|N| - 1$ arcs, so y_v is at most the length of a shortest directed sv-path.

Fact

Let $(v_0, v_1, \ldots, v_k)$ be a directed $v_0 v_k$-walk. Among all i, j for which $i < j$ and $v_i = v_j$, choose and j to minimize $j - i$. Then $(v_i, v_{i+1}, \ldots, v_j)$ is a directed cycle.
Proof left as an exercise.

Let W_v' be a directed sv- walk of length $\le y_v$ having as few arcs as possible. W_v is an example and there are finitely many walks of length $\le$ # arcs in W_v so W_ι' exists. If W_ι' is not a directed sv-path, it has a non-negative cycle we can excise to give a new walk, length $\le y_v$ and fewer arcs, a contradiction. So W_ι' is a directed sv-path of length $\le y_v$.
All paths have length $\ge y_v$ so the length of $W_v' = y_v$. ■

For (ii), let xz be an arc so that the final node weights satisfy $y_z > y_x + w_{xz}$. For each node v, let p_v be the node so that the time y_v was changed, it was changed because the arc $p_v v$ had $y_v^{(\text{then})} > y_{p_v}^{(\text{then})} + w_{p_v v}$. $\left(So\ y_z = y_{p_z}^{(\text{then})} + w_{p_z z}\right)$

At each stage of Bellman-Ford (for example, in iteration r, after checking the arc xz and, if $y_z < y_x + w_{xz}$, resetting $y_z = y_x + w_{xz}$), we can construct a directed graph with node set all x for which y_x is finite and, for each node x, the arc $p_x x$ where $p_x x$ is the most recent arc used to update y_x. Every node in this network, except possibly s, has precisely one arc pointing into this node.

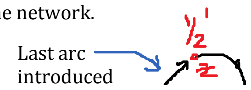
Observation

Suppose this network contains an sx-path P. Then the length of P is y_x
Length of P is the sum of the arclengths for the arcs in P.
For each arc xz of this network, since the last update of y_z, y_z has not changed. The length of w_{xz} ahs not changed. However, y_x might have changed; it can only get smaller.

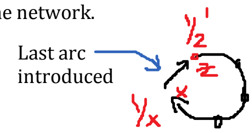
So each arc xz of P satisfies $y_z \ge y_x + w_{xz}$, or $w_{xz} \le y_z - y_x$
This implies $y_x - y_s \ge$ length of P
If $y_s < 0$, then we would have already found a negative cycle.
(Alternatively, arcs pointing in to s are irrelevant for shortest sx-paths and so may be deleted.)
We may assume $y_s = 0$ so $y_x \ge$ length(P)

At termination, we showed that the node-weight y_x is $\le$ length of every directed sx-path. If there is a directed sx-path in the final subnetwork, then this path is a shortest sx-path and its length is y_x

What happens if there is no directed sx-path in the final network? Then there is a directed cycle in the network.



What happens if there is no directed sx-path in the final network? Then there is a directed cycle in the network.



$y_x - y_s$ >length of P
$0 \geq$ length of the cycle
Before introducing xz, $y_z' > y_x + w_{xz}$ Length of directed zx-path in directed cycle is $\leq y_x - y_z'$
Length of directed cycle is length of this directed zx-path $+w_{xz}$
$\leq y_x - y_z' + w_{xz} < 0$

Overview

1) For $r = 0, 1, \dots, |N| - 1$, after iteration r, the node weight y_x $(< \infty)$ is the length of some directed sx-walk and every directed sx-walk having $\leq r$ arcs has length $\geq y_x$
2) If P is a directed sx-path in the terminal subnetwork, the P is a shortest sx-path in the original network and its length is y_x
3) If there is some x in terminal network with no directed sx-path, then there is a negative cycle.

st-flow

September-27-13 12:35 PM

st-flow

An st-flow in a network D with specified nodes s, t and capacity c_a on each arc a is a function x on the arcs of D so that

i) for each arc a, $0 \le x_a \le c_a$; and
ii) for each node v other than s, t,

$$\sum_{a,h(a)=u} x_a = \sum_{a,t(a)=u} x_a$$

Network has specified nodes s, t. Each arc e has a positive (nonnegative) capacity c_e. "water" to flow from s to t through the arcs, trying to get as much water out at t with no flow through any arc exceeding the capacity c_e of the arc e.

Flow constraints for each node v other than s and t,
sum of flows in to v = sum of flows out of v

$$\sum_{a,h(a)=v} x_a = \sum_{a,t(a)=v} x_a$$

The **value of the flow** is the net flow out from s

$$\sum_{a,t(a)=s} x_a - \sum_{a,h(a)=s} x_a$$

For any flow x, the value of the flow is

$$\sum_{a,t(a)=s} x_a - \sum_{a,h(a)=s} x_a \le \sum_{a,t(a)=s} c_a$$

so the sum of the capacities on the arcs leaving s is an upper bound on the maximum value of a flow.

Remark

Max st-flow is the linear program

$$\max \sum_{\text{tail}(a)=s} x_a - \sum_{\text{head}(a)=s} x_a$$

such that for every $v \ne s, t$,

$$\sum_{\text{head}(a)=v} x_v = \sum_{\text{tail}(a)=v} x_a$$

and for every arc a, $0 \le x_a \le c_a$

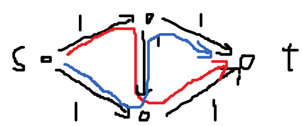
Augmenting Paths

Aiming for a Ford-Fulkerson max-flow algorithm & theorem.

Idea: from a given flow, find a new flow with greater value.

- there is a path from s to t that has positive excess capacity and so allows an augmentation.

Can we always find an augmenting path?



Given the red path, there is no path with excessive capacity, but we can use the blue path.