An algorithm for computing the fuzzy transitive closure of a bipolar weighted digraph

Mateja Šajna

March 12, 2014

1 Notation and terminology

We have a *bipolar weighted digraph* $D = (V, w_+, w_-)$, where:

- $V = \{1, 2, \ldots, n\}$ is the vertex set
- each arc is an ordered pair (s, t), with the arrow from vertex s to vertex t
- $w_+(s, t)$ is the weight of the positive arc (s, t), and $0 \leq w_+(s, t) \leq 1$
- $w_-(s, t)$ is the weight of the negative arc (s, t), and $0 \leq w_-(s, t) \leq 1$
- if the positive/negative arc (s, t) is absent from D, we set $w_+(s, t) = 0$ or $w_-(s, t) = 0$, respectively

The bipolar weighted digraph $D = (V, w_+, w_-)$ is represented by its matrices $A = (a_{st})_{s,t=1}^n$ and $B = (b_{st})_{s,t=1}^n$:

- A and B are of dimension $n \times n$;
- their entries are $a_{st} = w_+(s, t)$ and $b_{st} = w_-(s, t)$.

2 Fuzzy transitive closure

The fuzzy transitive closure of $D = (V, w)$ is the bipolar weighted digraph $D^* = (V, w_+^*, w_-^*)$, where (informally):

- $w_+^*(s, t)$ is the maximum weight of a minimal (s, t)-walk of positive sign
- $w_-^*(s, t)$ is (in absolute value) the maximum weight of a minimal (s, t)-walk of negative sign
- the sign of a walk is the product of signs of all arcs traversed by the walk

- the weight of a walk is (in absolute value) the minimum of the weights of the arcs of the walk
- a minimal (s,t)-walk of positive/negative sign is an (s,t)-walk not properly contained in an (s,t)-walk of the same sign

3 Input

Positive integer n and matrices $A = (a_{st})_{s,t=1}^{n}$ and $B = (b_{st})_{s,t=1}^{n}$ as described above.

4 Output

At the end of the algorithm, matrices $A = (a_{st})_{s,t=1}^{n}$ and $B = (b_{st})_{s,t=1}^{n}$, represent the fuzzy transitive closure of D. That is, $a_{st} = w_{+}^{*}(s,t)$ and $b_{st} = w_{-}^{*}(s,t)$ for all $s,t = 1,2,\ldots,n$.

5 Algorithm

```
procedure FuzzyTC(n, A, B)
begin
  Comment: compute the fuzzy transitive closure.
  for u = 1, 2, ..., n do        Comment: u is the new allowable vertex on the walk.
    for i = 1, 2 do        Comment: to allow for two traversals of vertex u.
      begin
  Comment: A' = (a'_st)_{s,t=1}^n and B' = (b'_st)_{s,t=1}^n will store the new entries of matrices A and
B, respectively.
        for s = 1, 2, ..., n do
          for t = 1, 2, ..., n do
            begin
              a'_st := max{a_st, min(a_su, a_ut), min(b_su, b_ut)}
              b'_st := max{b_st, min(a_su, b_ut), min(b_su, a_ut)}
            end
        for s = 1, 2, ..., n do        Comment: Update A and B.
          for t = 1, 2, ..., n do
            begin
              a_st := a'_st
              b_st := b'_st
            end
      end
  output A
  output B
```

Comment: now A contains the positive and B the absolute values of the negative weights of the fuzzy transitive closure.
end

6 Modification

If the input digraph contains no parallel arcs, of which one is positive and one negative (that is, there is no (s,t) with $w_+(s,t) > 0$ and $w_-(s,t) > 0$), then the input can be a single matrix $M = (m_{st})_{s,t=1}^n$ with entries

$$m_{st} = \begin{cases} w_+(s,t) & \text{if } w_+(s,t) > 0 \\ -w_-(s,t) & \text{if } w_-(s,t) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Input matrices A and B for procedure FuzzyTC can then be computed using the algorithm below.

procedure MatricesAB(n, M)

begin
Comment: copy all positive entries of M into matrix $A = (a_{st})_{s,t=1}^n$ and all negative ones (their absolute values) into matrix $B = (b_{st})_{s,t=1}^n$.

```
for s = 1, 2, ..., n do
    for t = 1, 2, ..., n do
        begin
            if m_st ≥ 0 then
                begin
                    a_st := m_st
                    b_st := 0
                end
            else
                begin
                    a_st := 0
                    b_st := −m_st
                end
        end
output A
output B
```

end