Extraction of components from a Social Network - a Graph Theoretic Approach

A dissertation submitted in partial fulfilment
of the requirements of M.Tech.(Computer Science)
degree at Indian Statistical Institute, Kolkata
by

Rajeev Tiwari

under the supervision of

Dr. Aditya Bagchi
Computer & Statistical Service Centre

Indian Statistical Institute
Kolkata-700 108.

July 17, 2002

Dedicated to my Mother

CERTIFICATE OF APPROVAL

This is to certify that the present dissertation titled *"Extraction of components from a Social Network - a Graph Theoretic Approach"* submitted by *Rajeev Tiwari* towards the partial fulfilment of the requirements for the award of M.Tech. in Computer Science degree at the *Indian Statistical Institute, Kolkata* is the bonafide work under my supervision,and it has not been submitted anywhere else for theaward of any other degree or diploma.

Bagchi 12.7.2002
(Supervisor)

ABandyopadhyay
(External Examiner)
12.7.2002

Aditya Bagchi
Computer & Statistical Service Centre
Indian Statistical Institute
Kolkata-700 108.

ACKNOWLEDGEMENT

I extend my sincere gratitude to *Dr. Aditya Bagchi*, Professor, CSSC, ISI, KOLKATA for his invaluable guidance, advice and continuous encouragement, without which it would not have been possible to produce this dissertation work.

I am extremely grateful to my friends and well-wishers who always stood by me when needed, especially during critical and painful moments.

RAJEEV TIWARI

Contents

1 INTRODUCTION

Social Network is a graph model that represents the relationship among the members of a community. For example, let us consider a village that has a number of households. If the members of a house communicate with another house, a directional edge is provided from the first to the second. If both the houses approach each other, then the edge is bidirectional. This way, the inter-relationship among the households of a village gives rise to a digraph. So a collection of villages would be represented by a collection of such digraphs. The social scientists are interested to extract certain properties of these digraphs in order to compare the social structures of the villages under study. The properties are extracted either from the adjacency matrix of the digraph representing a community or by a suitably designed graph algorithm. The properties may be enumerated as :

1. A community may have a few members who have no connection with the other members of the community. These nodes can be easily identified in the adjancency matrix. The corresponding row and column for any such "ISOLATOR" (an absolutely isolated member of a community) would have only zero entries signifying that both the indegree and outdegree for such a node are equal to zero. If a community has high percentage of ISOLATORs, it has possibly a new settlement.

2. Since all the members of a community may not communicate with one another, it is possible that a number of disjoint subgraphs may be present in the social network. These disjoint subgraphs are the "Weakly Connected Components (WCC)". The nodes of a WCC are somwhow connected but all the nodes may not be reachable from all others. Larger the size of the WCCs compared to the size of the actual community, higher is the cohesiveness of the community it is representing. An ideal society, would have a single WCC covering all the nodes of the community.

3. Each WCC within the network is considered as a group of mem-

bers within the community who are socially close to each other. Each WCC is a digraph by itself. If in a WCC, each node is reachable from each other node, it is a "Strongly Connected Component (SCC)". Many algorithms are available for finding SCCs in a graph [1,2,3,4,5]. However, in a social network, an SCC shows properties usually not present in other graphs. Since in a social network bidirectional links may be present between any two nodes, it is possible that an SCC or a WCC may have nested cycles. The standard cycle detection algorithms usually fail to identify these nested cycles.

This dissertation first identifies the ISOLATORs and then augments the adjancency matrix removing the rows and columns belonging to the ISOLATORs.

The augmented matrix is then used to find the WCCs present in the network.

The dissertation then offers an algorithm to find the different cycles (including the nested cycles) present in a social network.

Two households in a community may contact each other for more than one purpose. It may be for economic or social or for any other purpose. In such a case two nodes may be connected by more than one edge in the same direction. In the dissertation, the edges have been given different colours to represent the different type of edges. The algorithms have been suitably changed to find the network components under different edge colours.

The different components thus found from a social network are properly stored for future access. These information are later used by the social scientists to derive proper metrics to compare the different communities.

2 EARLIER WORK

Tarjan[3] has given a recursive algorithm based on depth first visit of nodes. It detects strongly connected components in a directed graph in $O(N+E)$ time.

Nuutila et. al.[2] have given two improved versions of Tarjan's algorithm to find the SCCs. The time complexity of the algorithm is $O(N+E)$.

Coremen et. al.[4] have presented linear-time $O(N+E)$ algorithm to find SCCs of a directed graph G,using two depth-first searches,one on G and another on G^T.

The above algorithms do not compute all the cycles of a directed graph.

Reingold et. al.[5] have given an algorithm to generate all the cycles in a digraph. The algorithm is applied to one SCC at a time. After generation of all cycles containing a node s, s is deleted from the graph. The overall procedure requires $O(\ (N+E)(\ \#\ \text{cycles} + 1\ )\ )$.

3 PRESENT PROBLEM

The graph G is defined by the set $\mathbf{G(N,E)}$,where $\mathbf{N = \{i|1 \leq i \leq n\}}$ is set of n nodes and $\mathbf{\{E = (i,j)|i,j \in N\}}$ is set of ordered pairs of nodes.An ordered pair (i, j) of nodes represents a directed edge in graph G.The node j is defined as an adjacent node to the node i.

In our case, the input file for a network is the adjacency matrix(nxn).The nodes are numbered serially from 1 to n.These are considered as node-id. The rows provide the outward connection from the node in the concerned row to the nodes along the row in the different columns.A matrix column will be 1 if there is a directed edge from the connected row to the concerned column.The element will be 0 otherwise.The diagonal elements are considered to be zero. If both (i,j) and (j,i) contain 0,nodes i and j are not directly connected in either direction.If a node k as row-no. has all the columns 0,then its out-degree=0.Again,a node k as column-no. has all the rows 0,then its in-degree=0.

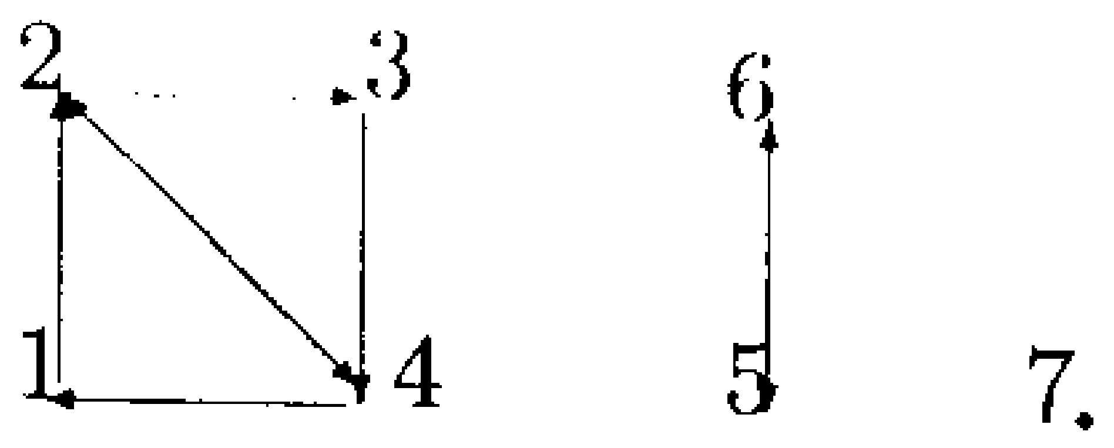
We have identified isolated nodes.These are not going to contribute in either cycle detection or any other path.A node with both in-degree and out-degree equal to zero is an isolated node.The count of isolated nodes has been taken and the percentage of isolated nodes in the graph has been reported.

From the adjacency matrix,the weekly connected components(WCCs) have been found out.A WCC is a sub graph structure in which all the nodes are reachable from all other nodes in it if edge directions are ignored.

The strongly connected component(SCC) is a subgraph structure in which all nodes are reachable from all other nodes in a digraph.An SCC may contain many cycles within it,including nested cycles.The present work is intended to list all the cycles of the given network.

The cycles have been fused to create hypernodes.The resultant graph structure is a Directed Acyclic Graph(DAG).

For example if the give network is as given below:



The adjacency matrix for the given network in the figure is shown below

$$\begin{matrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

The corresponding adjacency list is given below:

```
1  2  0
2  3  4  0
3  4  0
4  1  2  0
5  6  0
7  0
```

The different cycles in this network are as under:

Cycle-No.	size	contituent nodes
1	2	2 4
2	2	5 6
3	3	1 2 4
4	3	2 3 4
5	4	1 2 3 4

The present network has 5 cycles.Node no. 7 is an isolated node. After removal of isolated node, there will be two hypernodes as shown below:

Hypernode-id	Constituent nodes
h1	1 2 3 4
h2	5 6

The augmented adjacency matrix will be:

0 0
0 0

3.1 CYCLE DETECTION

The algorithm for the detection of cycles in a given graph is described below:

The graph G is defined by the set $\mathbf{G(N,E)}$,where $\mathbf{N} = \{\mathbf{i}|\mathbf{1} \leq \mathbf{i} \leq \mathbf{n}\}$ is set of n nodes and $\{\mathbf{E} = (\mathbf{i},\mathbf{j})|\mathbf{i},\mathbf{j} \in \mathbf{N}\}$ is set of ordered pairs of nodes.An ordered pair (i, j) of nodes represents a directed edge in graph G.The node j is defined as an adjacent node to the node i.

$temp\_visited(i)$: a flag which is made true when the node i has been temporarily visited.

$visited(i)$: a flag which is made true when the node i has been permanently visited.

C is an ordered sequence of nodes.

A concatenation operator is defined as follows:

.if $C = i\,j\,k$ then $C + p = i\,j\,k\,p$

A deletion operator is defined as follows:

.if $C = i\,j\,k\,p$

then $C - p = i\,j\,k$

A mask operator on the sequence C is defined as follows:

.if $C = i\,j\,k\,p\,q$

then $mask(2) = j\,k\,p\,q$,i.e. all the nodes in position i,$i < 2$ are deleted from the sequence. $m(i)$ specifies the position of node i in the sequence C,e.g.
.if $C = i\,j\,k$,then $m(k) = 3$
count is the number of nodes present in the sequence C.
.$i \in C$ is true when the node i is present in the sequence C.

ALGORITHM DESCRIPTION

Initial Conditions:

```
∀$i$
begin
        $Temp\_visited[i] = false$;
        $Visited[i] = false$;
end;
$C = empty$;

Procedure $Visit(i)$
begin
if $m(i) > 0$ then
        begin
        {$s \leftarrow m(i)$;
        find cycle($s$);
        end; }

  else
        begin
        $count \leftarrow count + 1$;
        $m(i) \leftarrow count$;
        $C \leftarrow C + i$;
        $for\, \forall k \in adj(i)$
        {$if\, notvisited[k]$
        $then\, visit(k)$;
}
          $C \leftarrow C - i$;
          $count \leftarrow count - 1$;
          $m(i) \leftarrow 0$;
          $Temp\_visited(i) \leftarrow true$;
          end;
end of $Visit$;

Procedure $Find\_Cycle(s)$;
begin
if for all $i \in C$
$temp\_visited(i) = true$;
then skip
else
output $mask(s)$ as a directed cycle;
end;
```

Main Algorithm :

```
$\forall i \in N$ do
begin
if $not\, temp\_visited[i]$ then
          begin
          $count \leftarrow 0$;
          $\forall j \in N\, do\, m(j) \leftarrow 0$;
          $Visit(i)$;
          end;
$\forall k \in N\ do$
if $Temp\_visited(k)$
then $Visited(k) \leftarrow true$;
end;
```

The Time Complexity of the algorithm is O((N+E)(Number of cycles in the graph)).

3.2 WCC GENERATION

The algorithm for the generation of Weakly Connected Components is presented below:

```
Procedure $WCCGEN(i)$
begin
if not $visited(i)$ then
        begin
        if $i$ not in $set(k)$ then add $i$ to $set(k)$;
        $\forall j$ such that $((i+1) \leq j \leq n)$ do
        if $(adj(i,j) = 1 or\, adj(j,i) = 1)\, and\, j \notin set(k)$ then
            begin
            add $j$ to $set(k)$;
            $WCCGEN(i)$;
            end;
        $visited(i)$ =true;
        end;
end.
```

Main Program(WCC)

```
begin
$k = 1$;
$\forall i(1 \leq i \leq n) visited(i) = false$;
      $\forall i$
        begin
        $WCCGEN(i)$;
        $k = k + 1$;
        end;
end.
```

3.3 EDGE LABEL CONSIDERATIONS

In case the edges have different labels(colours),the label may be perceived as the third dimension of the adjacency matrix,and as such there is no requirement of the separate algorithm for finding the cycles with edges having different possible labels.

4 CONCLUSION &FUTURE WORK

The present work thus presents an algorithm to find all the cycles, including nested ones in a given social network.The knowledge of every cycle present in the social network is very useful,especially to the social scientists.

The present algorithm can be made more efficient by parallelising it.

5 REFERENCES

1.A.V.,Aho,J.E. Hopecroft,and J.D. Ullman.*Data Structures and Algorithms.* Addison-Wesley,Reading,Mass.1983.

2.Esko Nuuutila,E. Soisalon-Saoininen.*On Finding the Strongly Connected Components in a Directed Graph,*Information Processing Letters,49(1),1994.

3.R.Tarjan.emphDepth first search and Linear Graph Algorithms. SIAM Journal of computing,1(2):146-160,June 1972.

4.T.Coremen et. al.*Introduction to Algorithms.*Prentice-Hall of India Private Ltd.,2000.

5.E.M.Reingold et.al.*Combinatorial Algorithms:Theory & Practice.*Prentice Hall,N.J.,1977.