Networks Structure and Dynamics: 12. Graph dynamics

Maximilien Danisch, Marwan Ghanem, Lionel Tabourier
LIP6 - CNRS and Sorbonne Université
first_name.last_name@lip6.fr

January 7, 2019

Plan

Outline

Context

Data

- Social Media;
- Motion;
- Infrastructure.

Important Entities

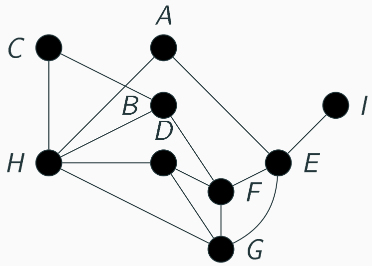
- Influencers;
- Super-spreaders;
- Critical points.

Degree Centrality

Degree Centrality:

$$degree(i) = \sum_{j=0}^{n} \mathcal{A}_{ij}.$$

Detect Influencers.



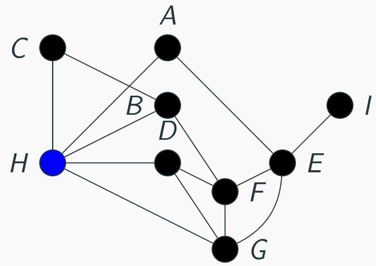
$$\begin{array}{c|ccccccccc} & A & B & C & D & E & F & G & H & I \\ \hline A & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ B & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ C & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ E & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ F & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ G & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ H & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

Degree Centrality

Degree Centrality:

$$degree(i) = \sum_{j=0}^{n} \mathcal{A}_{ij}.$$

Detect Influencers.



$$\begin{array}{c|ccccccccc} & A & B & C & D & E & F & G & H & I \\ \hline A & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ B & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ C & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ E & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ F & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ G & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ H & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

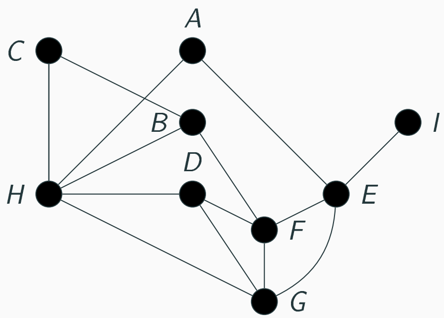
Closeness Centrality

Closeness Centrality:

$$Cc(u) = \sum_{v \neq u} \frac{1}{d(u,v)}.$$

$d(u,v)$: the distance between u and v.

Detect super-spreaders.



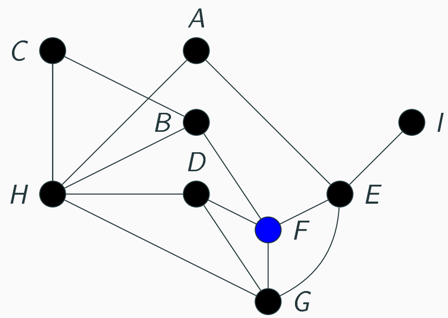
Closeness Centrality

Closeness Centrality:

$$Cc(u) = \sum_{v \neq u} \frac{1}{d(u,v)}.$$

$d(u,v)$: the distance between u and v.

Detect super-spreaders.

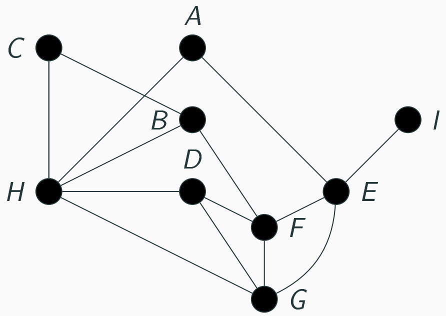


Betweenness Centrality

Betweenness Centrality:

$$Bc(u) = \sum_{s \neq u \neq t} \frac{\sigma_{st}(u)}{\sigma_{st}}.$$

σ_{st}: number of shortest paths between the nodes s and t.
$\sigma_{st}(u)$: number of shortest paths that go through u.



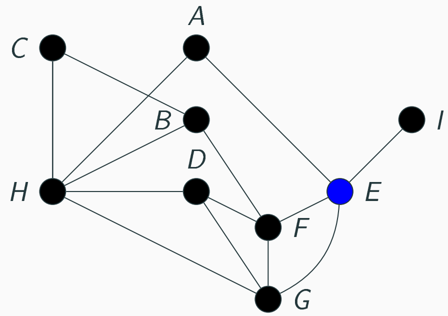
Detect critical points.

Betweenness Centrality

Betweenness Centrality:

$$Bc(u) = \sum_{s \neq u \neq t} \frac{\sigma_{st}(u)}{\sigma_{st}}.$$

σ_{st}: number of shortest paths between the nodes s and t.
$\sigma_{st}(u)$: number of shortest paths that go through u.



Detect critical points.

Motivation

But networks evolve ?

Motivation

Dynamic OF the network

Appearing and disappearing of

- nodes;
- links.

Dynamic ON the network

- diffusion of virus;
- sending messages.

Motivation

Dynamic OF the network

Appearing and disappearing of

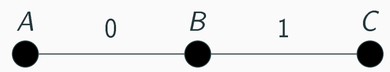
- nodes;
- links.

Dynamic ON the network

- diffusion of virus;
- sending messages.

What happens if we just ignore the dynamics ?

Example: Degree centrality

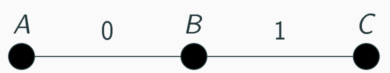


Questions

What is the degree of B ?

Is this meaningful ?

Example: Degree centrality



Questions

What is the degree of B ? 2

Is this meaningful ? Yes

Disadvantage

Omitting the dynamics of the graph:

- How long last nodes/links;
- Evolution;
- Temporal Paths;
- ...

Outline

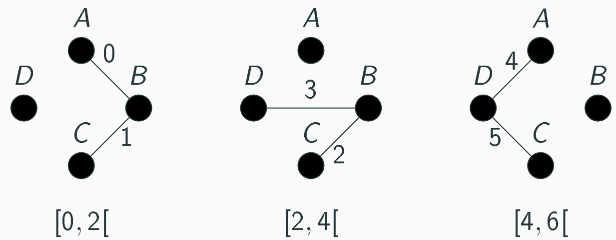
Introduction

Structural solutions

Temporal paths

Dynamic centralities

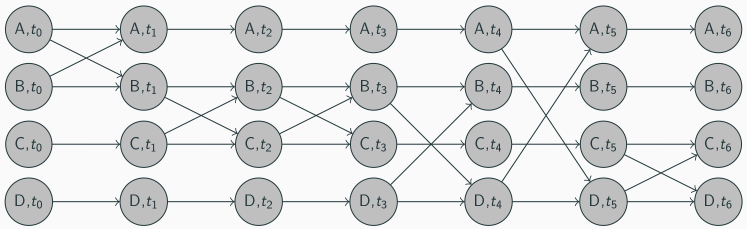
Snapshot



- Evolution is represented;
- Not all paths are represented;
- Some paths are wrong.

Uddin et al.14 - *in Social Computing (SocialCom), 2013*

Direct Acyclic graph



- Links of difference nature;
- Coverage Centrality:
  → fraction of shortest path that pass through the node.

Takaguchi et al.16 - *in The European Physical Journal B*

Outline

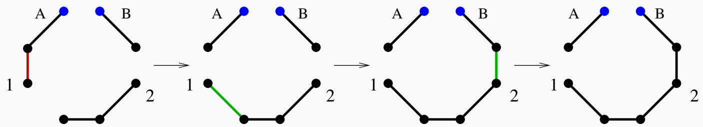
Introduction

Structural solutions

Temporal paths

Dynamic centralities

Temporal paths



Path from A to B.

No path from B to A.

Strong difference with the static version of distance.

Temporal path – definition

Définition intuitive

Succession of nodes $u_1, \ldots, u_k$ such that:

- there exists a link (u_1, u_2) at time t_1, (u_2, u_3) at time t_2, $\ldots$;
- $t_1 < t_2 < \ldots$;

Variants

Several variants according to different authors:

- The use of a link is immediate (nb of links one can use at a given time is then infinite);
- One needs δ to go through a link (given as a parameter);

More or less realistic according to the context.

More or less easy to compute.

Several definitions rely on paths, such as ...?

Distance

Distance

At least 3 natural definitions:

- the least number of hops;
- shortest in time to reach the target node;
- Fastest when transfer begins.

All the notions are useful depending on the context.

We will focus on the shortest in time to reach the target node.

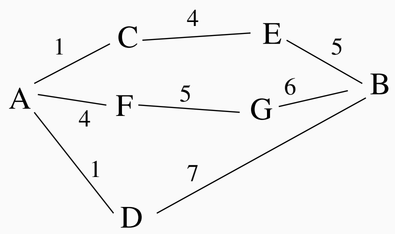
Distance – definition

Definition

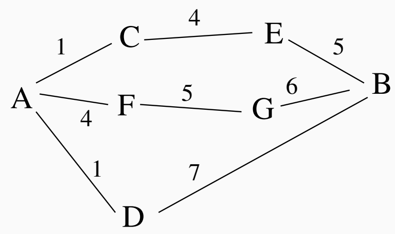
Let i, j be nodes, t a starting time.
Let t_a be the smallest time one needs to send a message from i to j (if possible).

Then the *temporal* distance from i to j at time t is: $t_a - t$.

Distance – examples



Distance – examples

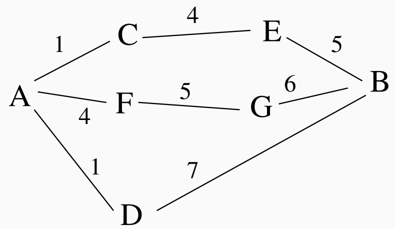


Distance de A à B

- Shortest in time to reach B: A – C – E – B;
- Least number of hops: A – D – B;
- Fastest when transfer begins: A – F – G – B.

The distance depends on the starting time!

Distance – examples



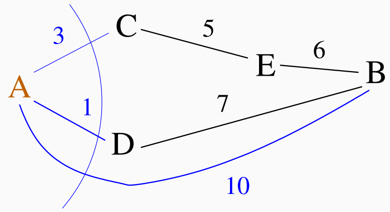
Distance de A à B

- Shortest in time to reach B: A – C – E – B;
- Least number of hops: A – D – B;
- Fastest when transfer begins: A – F – G – B.

The distance depends on the starting time!

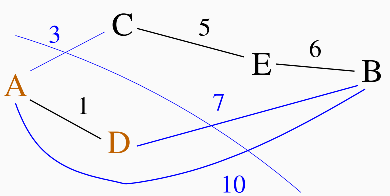
Exercise: ideas to obtain shortest (in time) from s to all nodes.

Distance – computation



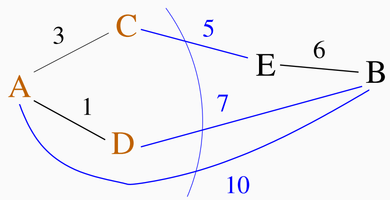
A 1

Distance – computation



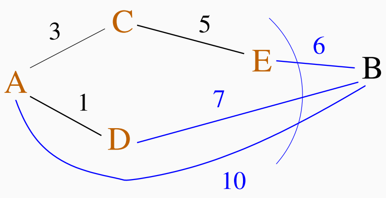
A 1
D 1

Distance – computation



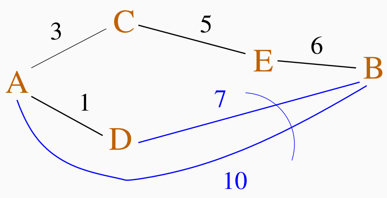
A 1
D 1
C 3

Distance – computation



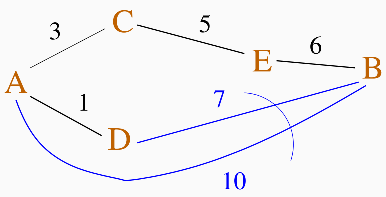
A 1
D 1
C 3
E 5

Distance – computation



A 1
D 1
C 3
E 5
B 6

Distance – computation



A 1
D 1
C 3
E 5
B 6

Exercise: Formalize the algorithm to obtain shortest (in time) from s to all nodes.

Complexity of the brute force approach

Brute force algorithm: transmission from node s to all others starting t_s

$Q = \{(s, x, t) \in E | t \geq t_s\}$
Mark s
$t_{cur} = t_s$

While $Q \neq \emptyset$:
    take the closest link to the border s.t. $t_{cur} \leq t$ (u, v, t)
    $t_{cur} = t$
    If v unmarked:
        mark v as reached at time t
        $\forall (v, x, t) \in E$, add (v, x, t) to Q

Complexity of the brute force approach

Brute force algorithm: transmission from node s to all others starting t_s

$Q = \{(s,x,t) \in E | t \geq t_s\}$
Mark s
$t_{cur} = t_s$

While $Q \neq \emptyset$:
    take the closest link to the border s.t. $t_{cur} \leq t$ (u,v,t)
    $t_{cur} = t$
    If v unmarked:
        mark v as reached at time t
        $\forall (v,x,t) \in E$, add (v,x,t) to Q

Complexity

While : look at (almost) all links: m

maintaining Q : $\sim \log m$

$\sim m \log m$ steps

Complexity of the brute force approach

Brute force algorithm: transmission from node s to all others starting t_s

$Q = \{(s,x,t) \in E | t \geq t_s\}$
Mark s
$t_{cur} = t_s$

While $Q \neq \emptyset$:
    take the closest link to the border s.t. $t_{cur} \leq t$ (u,v,t)
    $t_{cur} = t$
    If v unmarked:
        mark v as reached at time t
        $\forall (v,x,t) \in E$, add (v,x,t) to Q

Complexity

While : look at (almost) all links: m

maintaining Q : $\sim \log m$

$\sim m \log m$ steps
... and for any node to any node?

Complexity of the brute force approach

Brute force algorithm: transmission from node s to all others starting t_s

$Q = \{(s,x,t) \in E | t \geq t_s\}$
Mark s
$t_{cur} = t_s$

While $Q \neq \emptyset$:
    take the closest link to the border s.t. $t_{cur} \leq t$ (u,v,t)
    $t_{cur} = t$
    If v unmarked:
        mark v as reached at time t
        $\forall (v,x,t) \in E$, add (v,x,t) to Q

Complexity

While : look at (almost) all links: m

maintaining Q : $\sim \log m$

$\sim m \log m$ steps
... and for any node to any node? ... and for any node to any node starting at any time?

Outline

Introduction

Structural solutions

Temporal paths

Dynamic centralities

Betweenness centralities

Let G be a graph and v, s and t be nodes of G.

We call *betweenness centralities* the value:

$$BC(v) = \sum_{s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

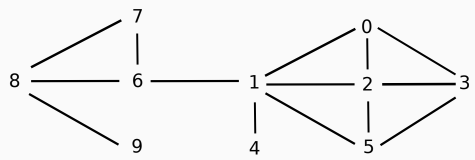
with:

- σ_{st} : nb of shortest paths between s and t;
- $\sigma_{st}(v)$: nb of shortest paths between s and t going through v.

Meaning

Captures the importance of a node v in a graph as a relay for diffusing information:

- propagation of signals/messages/virus;
- connectivity of the network.



Exercise : compute $BC(0)$, $BC(1)$ and $BC(4)$.

Algorithm

One needs to know **all** shortest paths between s and t.

Algorithm

One needs to know **all** shortest paths between s and t.

- BFS: 1 shortest path
- Needs to modify the algorithm. Idea:

Exercise : Compute the DAG of all shortest paths starting from 3 in the previous graph.

Algorithm

One needs to know **all** shortest paths between s and t.

- BFS: 1 shortest path
- Needs to modify the algorithm. Idea:
  - Compute a spanning DAG (Directed Acyclic Graph) ;
  - Rely on the computation of the distances;
  - Use the depth of a node (distance to the root) in order to decide if a node is already part of the DAG or not.

Exercise : Compute the DAG of all shortest paths starting from 3 in the previous graph.

Algorithm

Exercice :

- Propose an algorithm computing the DAG of all shortest paths, given a graph G and a node s;
- Given a DAG of shortest paths, propose a formula allowing to compute, for every node v, the couple (dn, up) where:
  - dn : nb of downward paths of v;
  - up : nb of upward paths of v.
- Deduce the number of shortest paths starting from s and going through v;
- Apply the algorithm to the DAG starting from 3 in the previous graph;
- Propose the final algorithm enabling to compute the betweenness centrality of a given node.

Betweenness centrality

Which dynamic version for centrality?

Dynamic centrality – several propositions

Evolution of the standard centrality

Compute the Betweenness centrality for each time intervals

- depends on the size of the window;
- do not take into account realistic communications in most of the cases.

Direct acyclic representation

Compute the Betweenness centrality for each node

- Expensive.

Extension to temporal paths

For a node i, compute the fraction of shortest temporal paths going through i.

- Temporal paths depends strongly on starting time.

Temporal Betweenness Centrality

Nicosia et al. - *in* Temporal Networks, *2013*

Take into account the waiting time on the nodes:

If one has a unique shortest time from i to j :
$i \longrightarrow k \longrightarrow j$

the importance of k depends on the time the message "spends" on k

Temporal Betweenness Centrality – definition

Nicosia et al. - *in* Temporal Networks, *2013*

Take into account the waiting time on the nodes:

$$C_i(t_m) = \frac{1}{(n-1)(n-2)} \sum_{j \neq i} \sum_{k \neq i,j} \frac{U(i, t_m, j, k)}{\sigma_{jk}}$$

- $U(i, t_m, j, k)$: nb of shortests temporal paths from j to k such that one uses i at a time $\leq t_m$;
- σ_{jk} : nb of shortest temporal paths from j to k.

Average centrality: average over all time instants

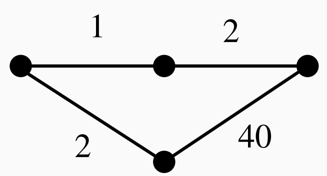
Temporal Betweenness Centrality – drawbacks

- All paths start at the initial time!
- Only the average value is studied.

Mediation

Tang et al. - *in* Temporal Networks, *2013*

Idea
If k is on a shortest temporal path between i and j
its importance depends on the second shortest temporal path.



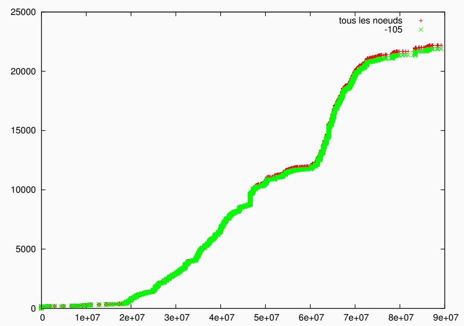
Mediation – in practice

Principe

- Compute all distances between all pairs of nodes;
- Suppress the node i;
- Compute again all distances
  Difference : importance of node i.

Mediation - example

Cumulative distribution of distances



Need to take into account paths that start at all instants

Importance of a node – other propositions

Alternatives exist

- Closeness centralities and extensions
  Time Evolution of the Importance of Nodes in dynamic Networks, Magnien & Tarissan - *in* Asonam*, 2015*
- Coverage centrality
  Coverage centralities for temporal networks, Takaguchi *et. al - in The European Physical Journal B, 2015*
- . . .

Conclusion

Properties defined for static networks are insufficient to describe dynamic networks

In this course:

- Properties related to temporal paths

Other properties

- Temporal patterns;
- Duration of nodes/links;
- Resilience of links;
- Dynamic communities;
- . . .