Detecting structural changes and command hierarchies in dynamic social networks

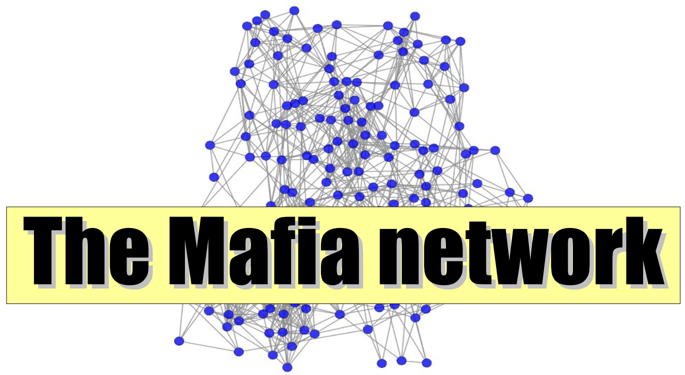
Authors: Romain Bourqui °
Frédéric Gilbert *
Paolo Simonetto *
Faraz Zaidi *
Umang Sharan ⫠
Fabien Jourdan ‡

* LaBRI, University of Bordeaux
* INRIA Bordeaux Sud-Ouest
° Eindhoven University
⫠ Purdue University
‡ INRA Toulouse

What is this for?

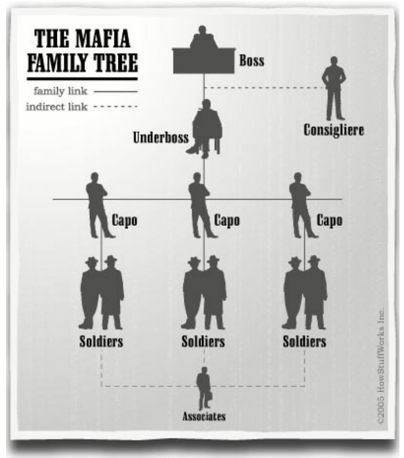
The evolution of social networks is often depicted with dynamic graphs.

How can we study dynamic graphs to detect:



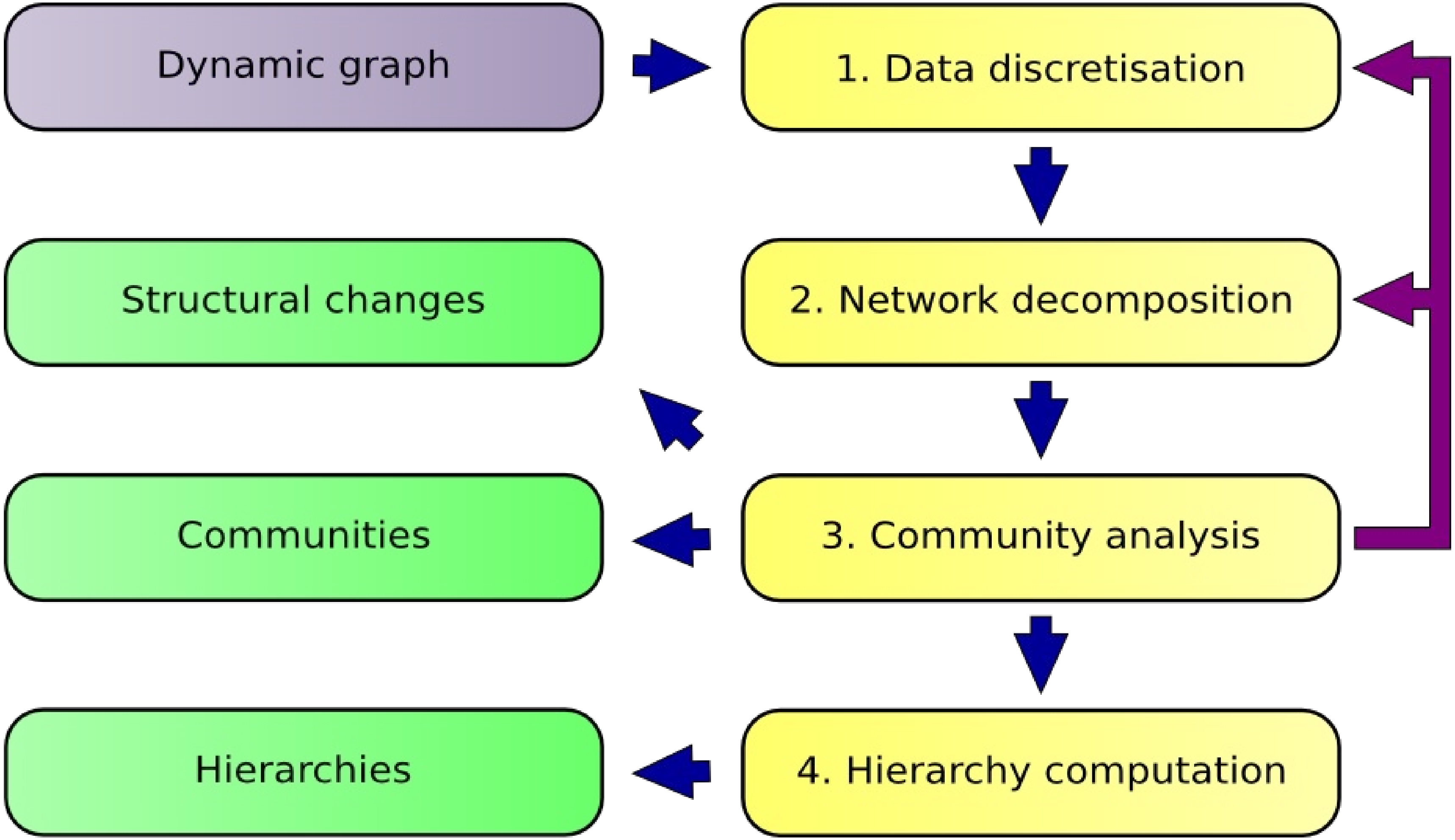
network structural changes,

communities,



command hierarchies?

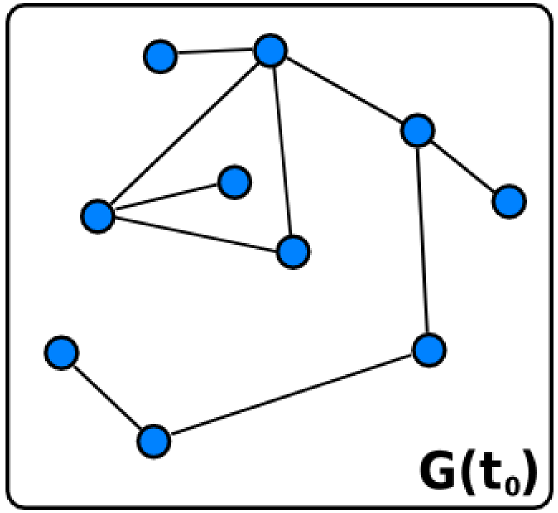
The framework

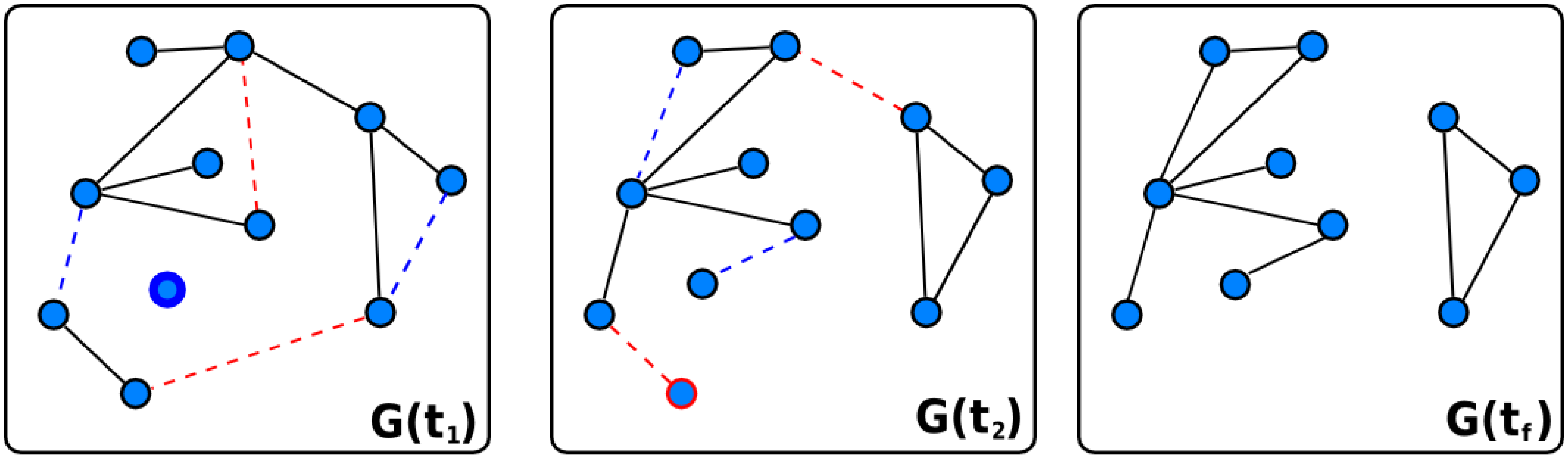


The input: dynamic graphs

A dynamic graph is a graph whose structure changes in function of the time.

Both nodes and edges can appear or disappear.

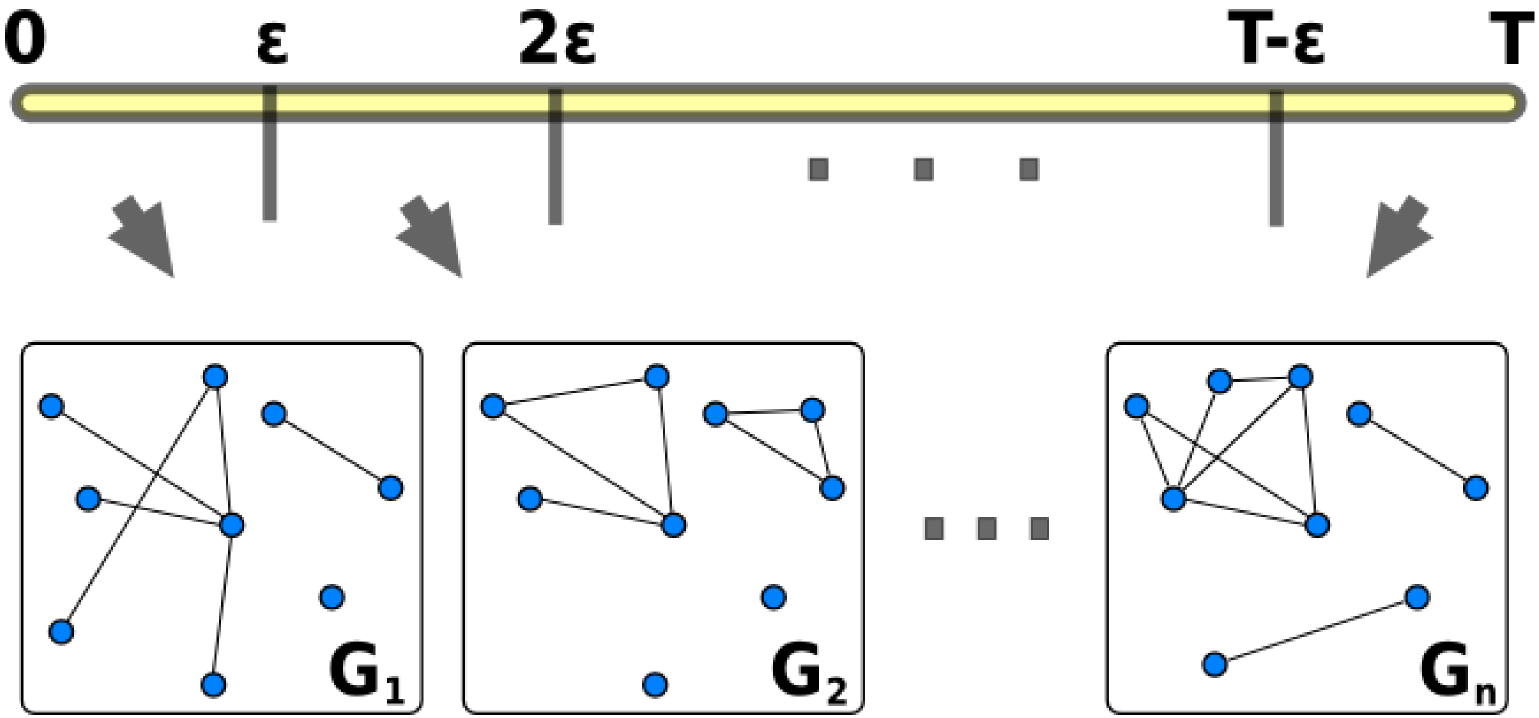




Step 1: Dynamic graph discretisation

The interval of time where the dynamic graph is defined, [0,T], is divided in periods of duration ε.

For each period [t_1,t_2], we construct a static graph by merging all the elements that are present in the dynamic graph at any time $t \in [t_1,t_2]$.



Step 2: Static graph clustering – Strength metric

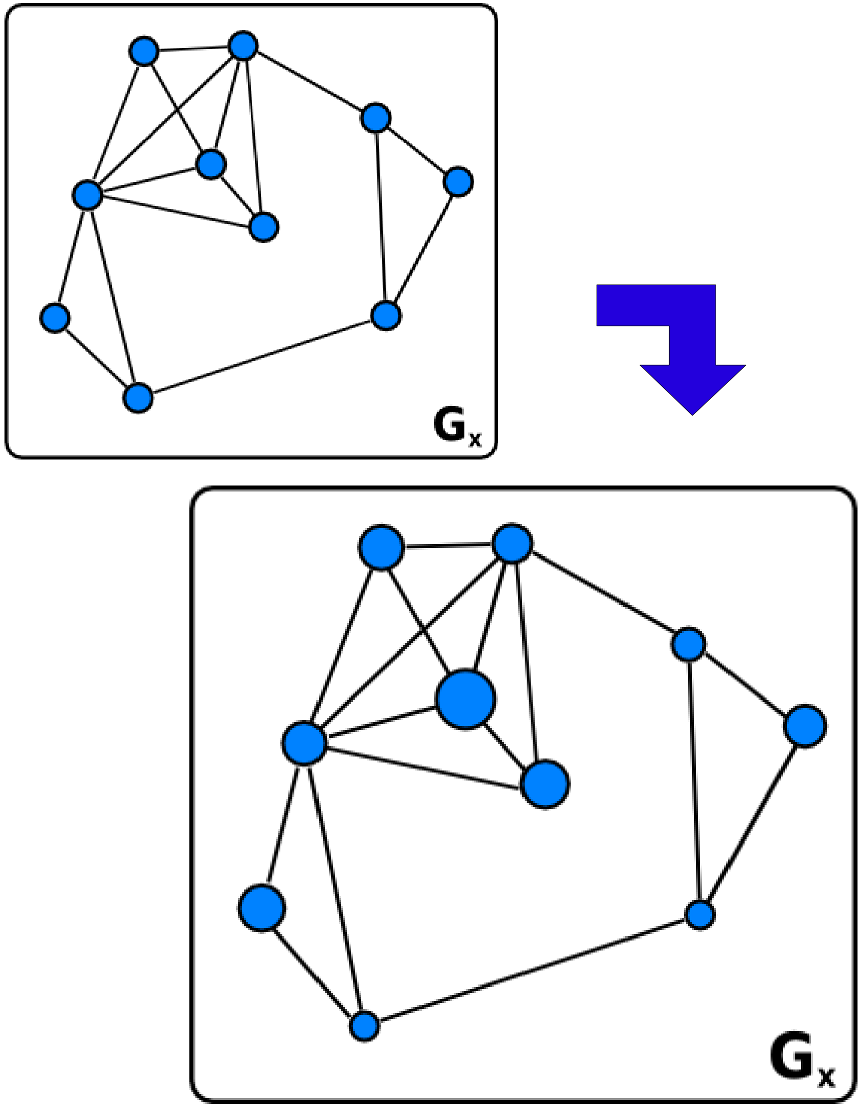
Each static graph *G* is weighted using the Strength metric.

Strength metric:

- is defined for edges.
- evaluate the interconnections between the neighborhoods of an edge's incident nodes.

D. Auber, Y. Chiricota, F. Jourdan, and G. Melançon. Multiscale Visualization of Small-World Networks. *IEEE Information Visualization Symposium*, pages 75–81, Seattle, USA, 2003. IEEE Computer Press.

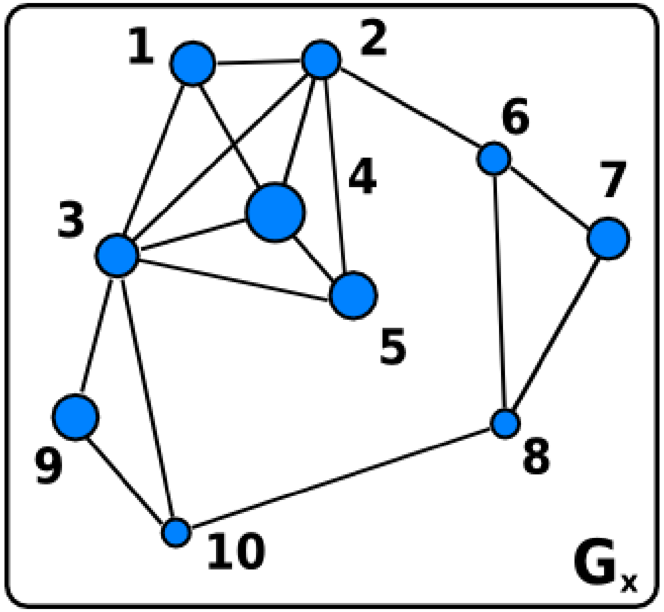
We assign a strength value for each node calculating the average metric value of the incident edges.



Step 2: Static graph clustering – Cluster centres

Our clustering technique is based on the identification of the central element of a cluster.

1. Sort the graph nodes according to the strength metric.
2. Select the best node of the list and promote it to cluster centre.
3. Eliminate the cluster centre and its neighbours from the list.
4. Iterate the process from 2 until no more nodes are in the list.



4	**5**	**1**	**7**	**9**	**3**	**2**	**6**	**10**	**8**
4	~~5~~	~~1~~	**7**	**9**	~~3~~	~~2~~	**6**	**10**	**8**
4	5	1	**7**	**9**	3	2	~~6~~	**10**	~~8~~
4	5	1	7	**9**	3	2	6	~~10~~	8

Cluster centres: 4 7 9

Step 2: Static graph clustering – Cluster detection

Clusters are detected investigating the neighbourhood of the centres.

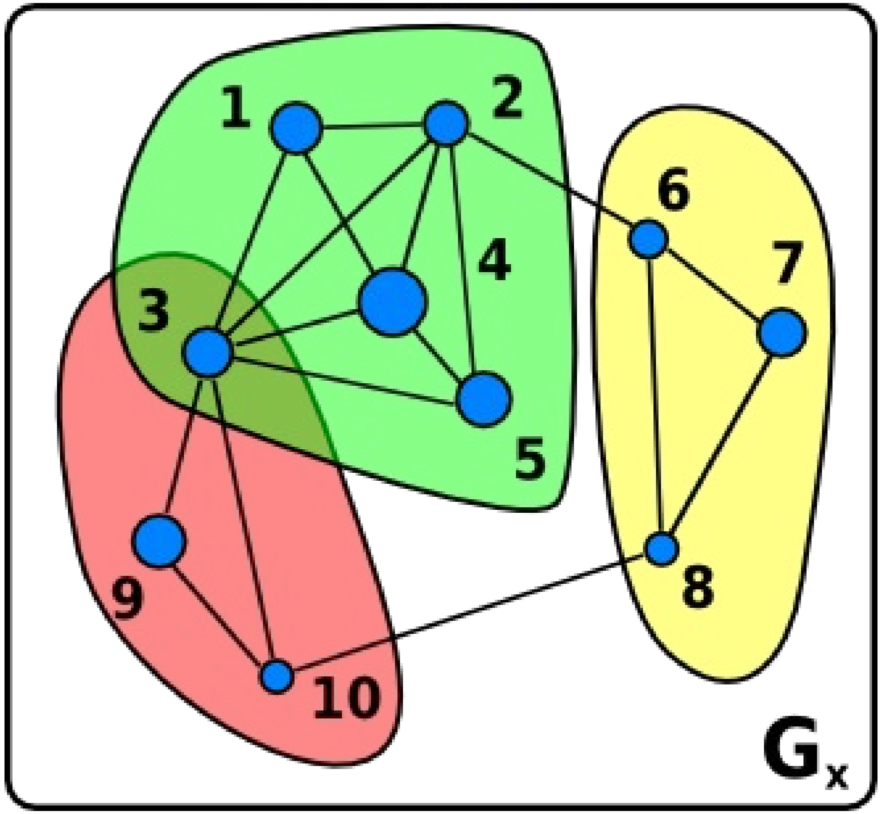
For each cluster centre, we define a set of nodes *S*. This contains the centre and all neighbours with metric greater than a threshold τ.

The clusters are the subgraph induced by the sets *S*. They might overlap between them.

Each static graph *G* is clustered using *m* values of the threshold *τ*.

This gives a total of *n***m* clusterings.

Cluster centers: 4 7 9



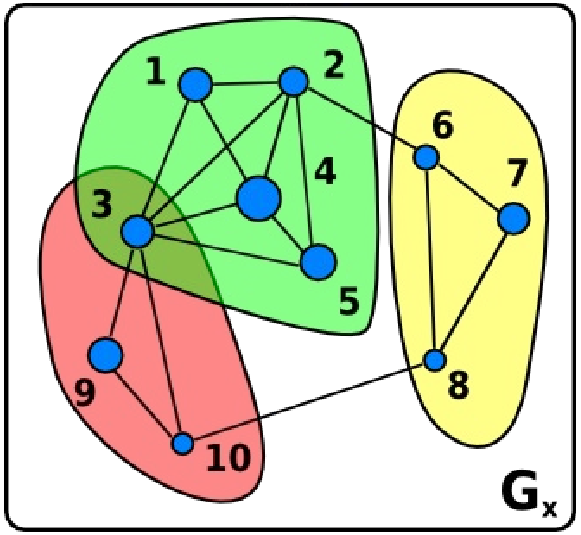
Step 3: Community analysis – Clustering similarity

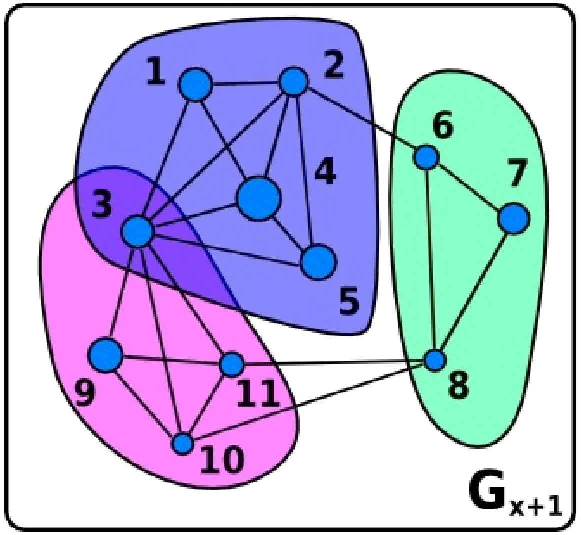
Communities and structural changes are detected analysing the similarity between clusters and clusterings.

Two clusters are similar if one contains a high ratio of the elements of the other cluster, and vice-versa.

Two clusterings are similar if one contains a similar cluster for all the clusters of the other, and vice-versa.

Formulas and more details on the metrics can be found in the paper.

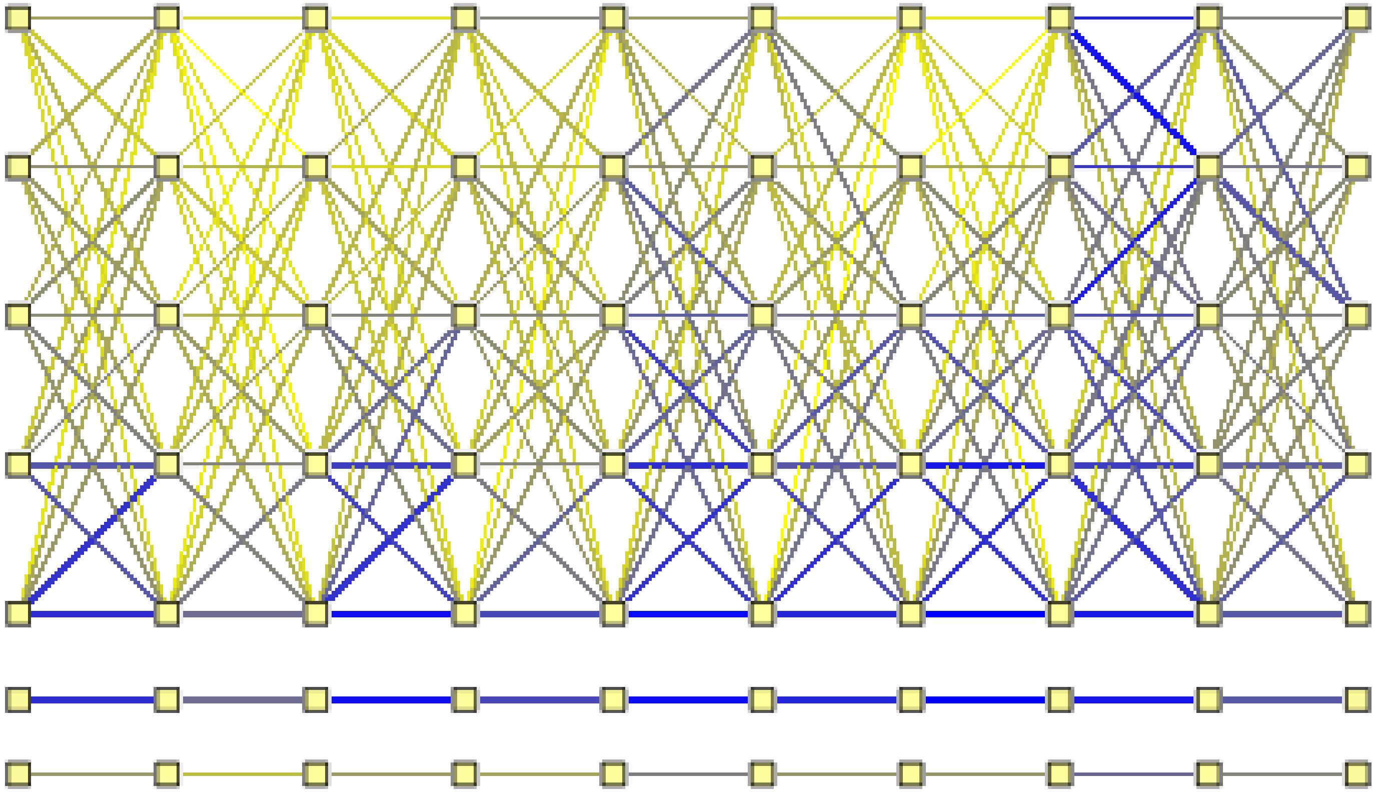




Step 3: Community analysis – Structural changes

The similarity metric is computed for each pair of consecutive clustering, and the results are visualised in graph form.

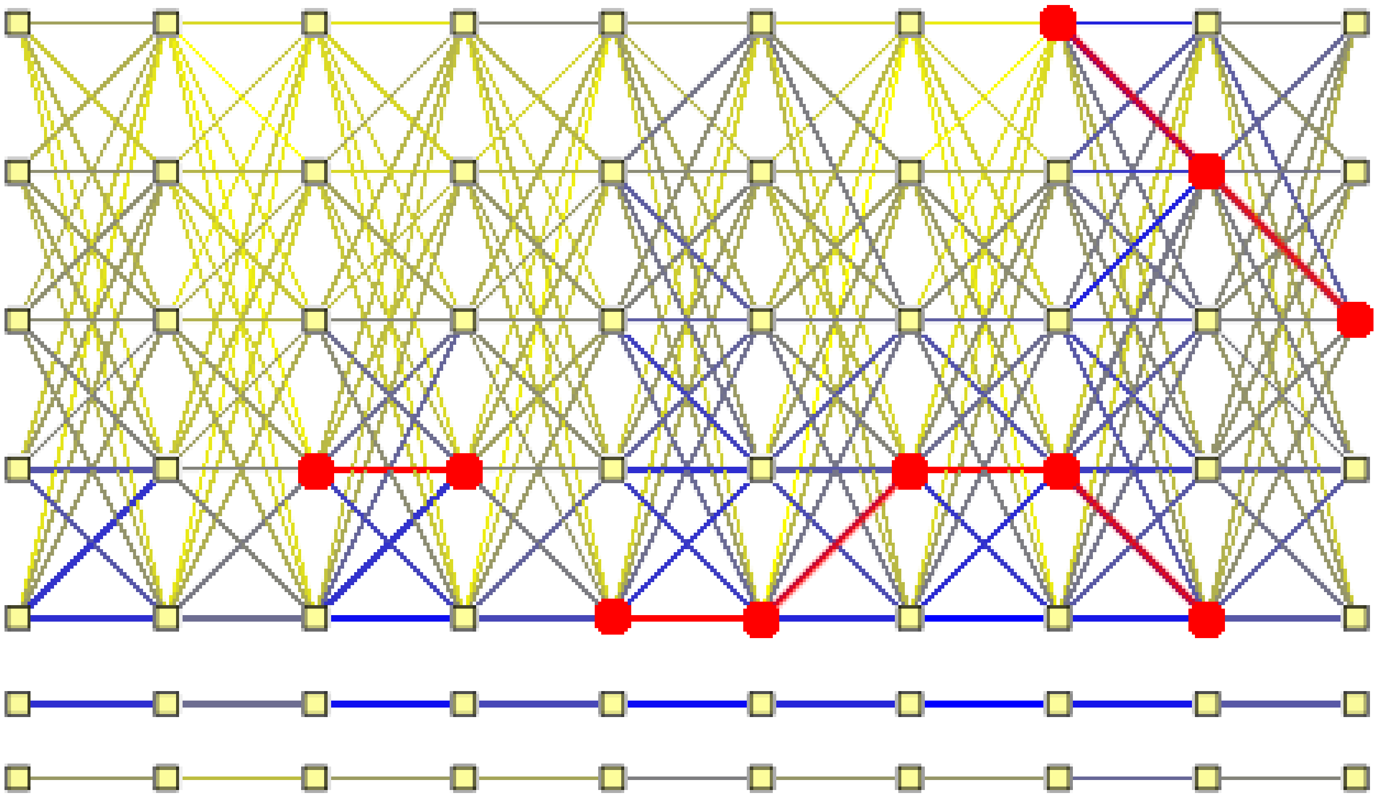
Evolution inertia helps to detect the most appropriate clustering for G_x.



Step 3: Community analysis – Consensus communities

Consensus communities can be calculated choosing paths in the graph and combining the best matching clusters of different clusterings.

Clusters can be combined in many ways (union, intersection, ...).



Step 3: The feedback loop

The matching graph gives important information about the parameters chosen in the previous steps.

Very few good matchings?

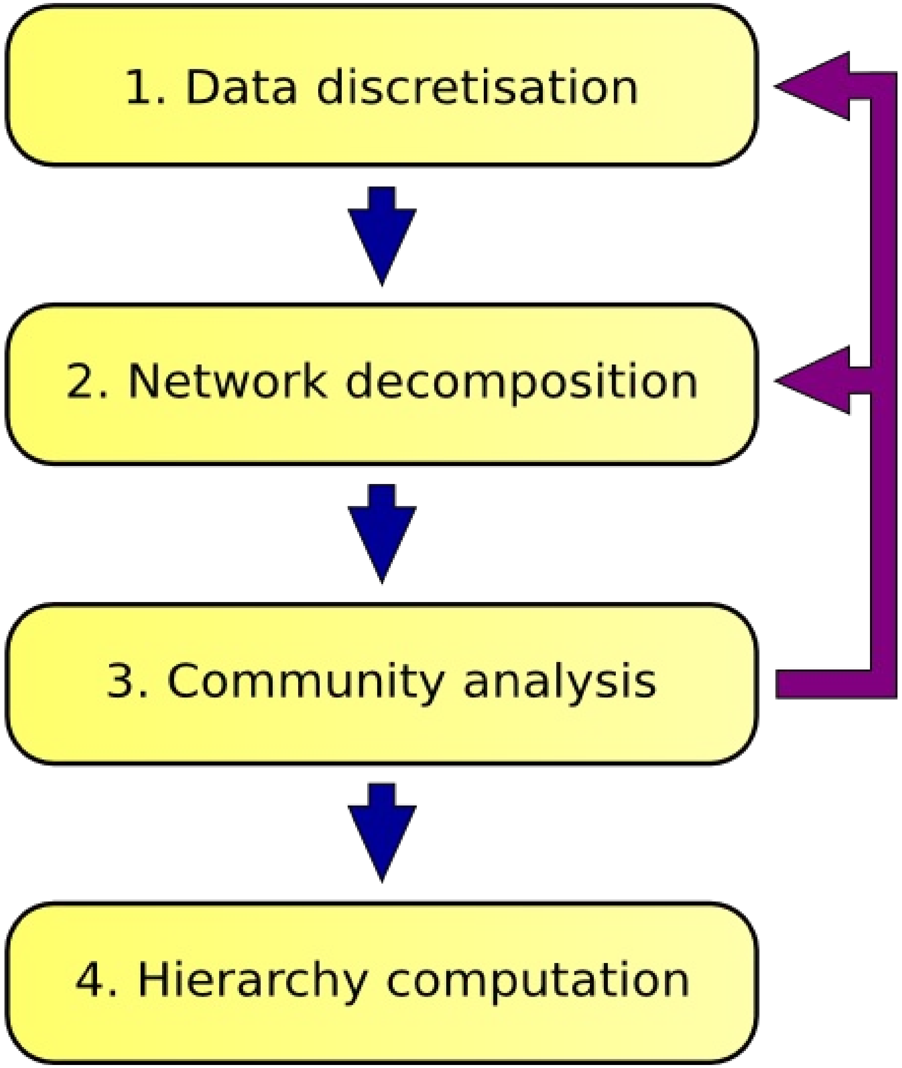
- the discretisation interval might be too big.

The good matchings are not evenly distributed?

- It might be useful to reconsider the values chosen for τ.

Nothing works?

- It might be time for a break.



Step 4: Hierarchies – Efficiency and importance

Efficiency is a metric on *graphs*.
It evaluates if there are many or few different shortest paths between its nodes.

Importance is a metric on *nodes*.
It evaluates how the efficiency changes by removing the given node from the graph.

Latora & Marchiori. How Science of Complex Networks can help in developing Strategy against Terrorism. *Chaos, Solutions and Fractals*, 2004.

$$i, j \in V, \quad i \neq j$$

$$d_{i,j} = |SP(i,j)|$$

$$\epsilon_{i,j} = \frac{1}{d_{i,j}}$$

$$Eff(G) = \frac{\sum_{i \neq j \in V} \epsilon_{i,j}}{|V| \times (|V| - 1)}$$

$$I(i) = Eff(G) - Eff(G \setminus i)$$

Step 4: Hierarchies – What importance shows?

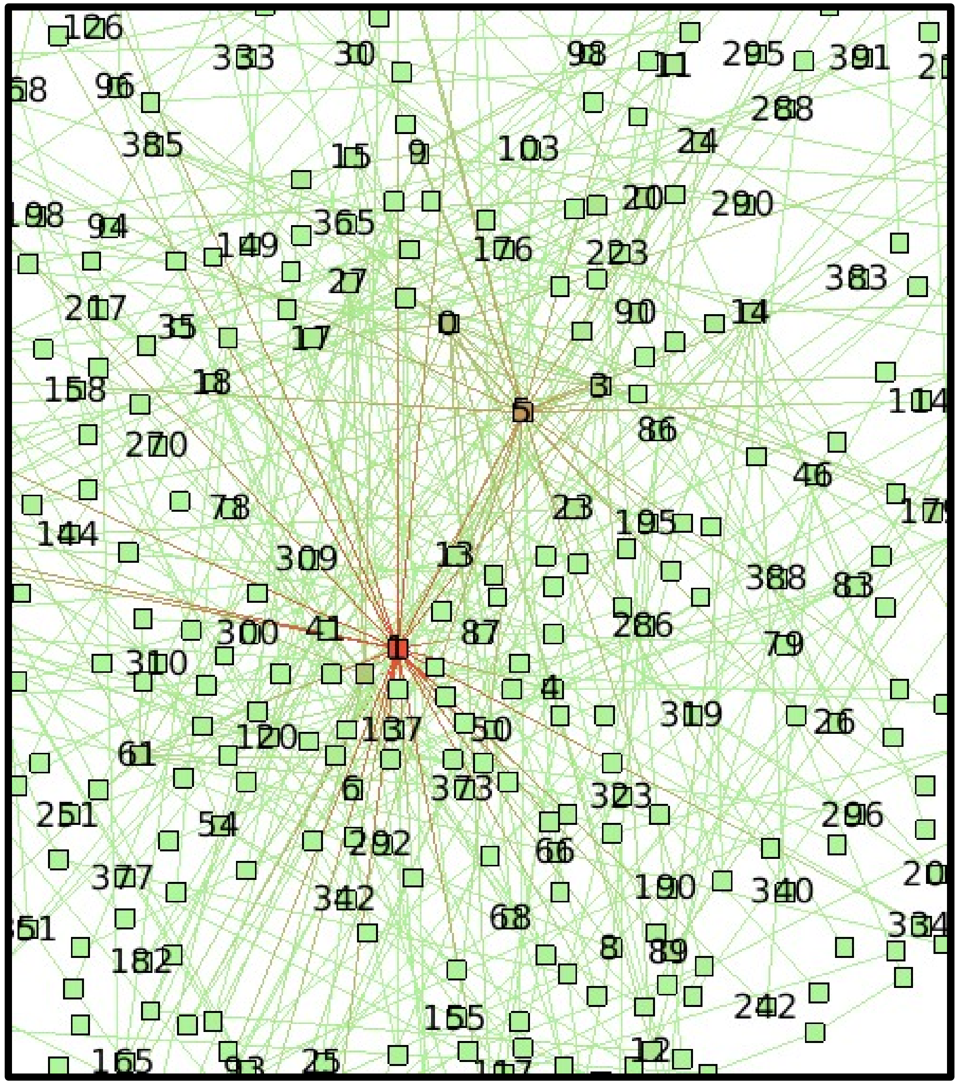
The importance of the nodes have been computed on data from the Castlelano/Vidro dataset.

Importantance is encoded in a colour scale from green to red.

There are two kind of nodes:

- **gatekeepers** with high Importance values
- **followers** and **leader**(**s**) with low importance values

Leaders try to hide themselves among followers.



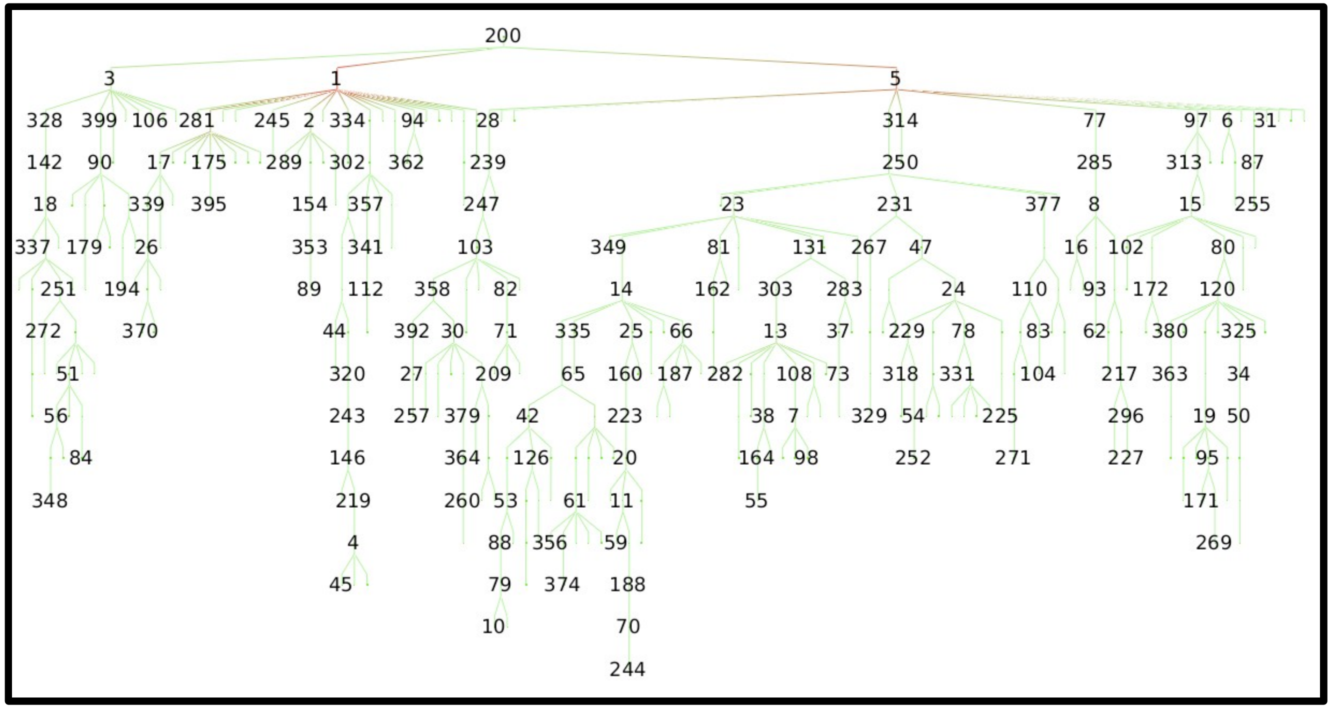
Step 4: Hierarchies – Detecting the structure

Edges are weighted with the difference between the importance of the nodes.

A spanning tree is computed with the Kruskal's MST algorithm.

The leader(s) are detected in intersection between the gatekeepers' neighbours.

The tree can now be rooted to obtain the command hierarchy.



Conclusions

The framework presented allows the analysis of dynamic social networks to obtain information on their structure.

Input:

- Dynamic graph describing the evolution of the social network

Output:

- Description of the network structural changes
- Consensus communities
- Command hierarchy

The framework have been tested on the Castelano/Vidro dataset. Studying the phone call data of the dataset, we succesfully identified:

- episodes of major structural changes of the organisation,
- the influence hierarchy of the family Castelano/Vidro.

Thank you for your attention.