M2 – NSD (Practical Work 7 - Sessions 12-13-14)

Graph Dynamics

Maximilien Danisch, Marwan Ghanem, Lionel Tabourier
(adapted from Fabien Tarissan)

1 Part 1 - Temporal interaction without duration

The purpose of part 1 is to understand the effect of omitting temporal information. How the nodes importance differs from a static to a temporal analysis. To do so, we will test the strategies on real-world datasets. While it is preferable to code the algorithms from scratch, it is possible to use existing algorithms from the available packages.

1.1 Preliminaries

Download the following datasets : `https://www-complexnetworks.lip6.fr/~ghanem/datasets.gz`.
Those datasets consists of lines of tuples describing the interactions between the different entities. The format is :
`n1 n2 t`
where `n1` and `n2` are the ids of the two nodes involved in the interaction at time `t`.
For the sake of exemplifying, here is a small datasets involving 4 nodes during 10 units of times :

```
0 2 0
1 3 1
0 1 5
0 3 9
```

Exercise 1 — *Properties of the original networks*
For each datasets, compute the number of nodes, the number of links and the total duration of the dataset.

1.2 Normalization

Those datasets don't start at time 0, which makes the comparison difficult. We will homogenize the interaction times, by making all the experiments start at time 0.

Exercise 2 Write a program that normalizes all the files such that they all start at time 0. The files are ordered by time of contact in an increasing order. After normalizing, the files should remain ordered in the same manner.

1.3 Temporal Analysis

1.3.1 Inter-contact

We call *inter-contact duration time* the time that separates two contacts involving the same pair of nodes.

Exercise 3 Write a program that computes the inter-contact duration time associated to each observed contact.

Exercise 4 Using gnuplot, compute and display the distribution of the inter-contact duration times. Comment the plots.

1.3.2 Distance

As the nature of the dataset changes, the (temporal) distances between nodes vary. A dense dataset with high activity, will have short distances between the nodes, while a dataset with low activity will have long distances between the nodes.

Exercise 5 Using the lecture slides, implement the brute force approach. Analyze each dataset by studying the distribution of distances at several instants, justify the choice of these instants.

1.4 Centralities

1.4.1 Snapshot

Exercise 6 Implement the following centralities for a static graph : Degree centrality, Closeness centrality, Betweenness centrality

Exercise 7 Write a program that for a given dataset, will generate several snapshots, each snapshot lasts d seconds.

Exercise 8 Write a program that for a give dataset and a given node id, computes the previous centralities. The output should be in the following format :

```
t1 degree betweenness closeness
t2 degree betweenness closeness
t3 degree betweenness closeness
```

For a single dataset and a randomly selected node id, run the previous program for different snapshot sizes. Using gnuplot, plot the different outputs and comment the effect of changing the snapshot size. Justify the choice of the snapshot sizes.

1.4.2 Directed acyclic graph (Optional)

Exercise 9 Write a program that for a given dataset, will generate the graph in the form of a Directed acyclic graph.

Exercise 10 Using the previous program, run the three centralities for a given node using the smallest dataset of the four.

1.4.3 Temporal paths (Optional)

Exercise 11 Implement the algorithm proposed by Tang *et al* in Temporal Networks.

1.4.4 Comparison (Optional)

Exercise 12 Propose a method to compare the different methods proposed above. The proposed method should show how for a single node the importance is perceived by each method.

2 Part 2 - Temporal interaction with duration

The purpose of part 2 is to do some simple manipulations on data with interaction durations.

2.1 Preliminaries

Download the dataset RollerNet and Infocom06 from `http://lioneltabourier.fr/enseignements.html` (be careful, the RollerNet dataset is not identical to the one studied in part 1).
Those datasets consists of lines of tuples describing the interactions between the different entities. The format is :
`n1 n2 t1 t2`
where `n1` and `n2` are the ids of the two nodes involved in the interaction which starts at time `t1` and ends at time `t2`. It is worth noticing that the contacts are undirected and that, by convention, `n1` < `n2`.

For the sake of exemplifying, here is a small dataset involving 4 nodes during 10 units of time :

```
1 2 126 128
1 3 125 128
1 4 123 128
2 3 121 121
2 4 120 121
2 4 124 126
3 4 122 128
```

2.2 Normalization

Exercise 13 Write a program that normalizes all the files such that they all start at time 0.

On the former example, it yields :

```
1 2 6 8
1 3 5 8
1 4 3 8
2 3 1 1
2 4 0 1
2 4 4 6
3 4 2 8
```

Contrary to the previous section, those files have been sorted by identifier in increasing order. Sometimes, it is more useful to use a version of the dataset sorted by time and in which the information related to starting and ending a contact is split. Therefore, a line :
`n1 n2 ts te`
will generate two lines
`ts n1 n2 C`
and
`te+1 n1 n2 S`
where the fourth column indicates is a character (C or S) that indicates whether the event is referring the the starting time or ending time of a contact (or, in other word, whether it is a creation or a suppression of a link).
Remark : Be careful that the suppression of the link occurs at time `te+1`, not `te`.

In our small example, this results in :

```
0 2 4 C
1 2 3 C
2 2 3 S
```

```
2 2 4 S
2 3 4 C
3 1 4 C
4 2 4 C
5 1 3 C
6 1 2 C
7 2 4 S
9 1 2 S
9 1 3 S
9 1 4 S
9 3 4 S
```

Exercise 14 Write a program that generates such a version of the data file.

If you want to do it using bash scripting, let us recall that the command `sort -n -k`x, x `file` allows to sort the file `file` according to the numerical increasing order of the values contained in column x.

2.3 Analysis

This part is dedicated to the analysis of different properties of contact networks.

2.3.1 Inter-contact

Again, we call *inter-contact duration time* the time that separates two contacts involving the same pair of nodes For the second contact between 2 and 4 in the example, the inter-contact duration time is 2.

Exercise 15 Write a program that computes the inter-contact duration time associated to each observed contact.

Exercise 16 Compute and display the distribution of the inter-contact duration times.

2.3.2 Average degree

We focus now on the evolution of the average degree of the nodes in the network.

Exercise 17 Write a program that computes the average degree at each instant. Is it mandatory to compute the degree of each nodes to answer the question ?

Exercise 18 Using gnuplot, display the evolution of the average degree over time for the datasets RollerNet and Infocom06. Comment the plots.

2.4 Creation and deletion of links

We focus now on the dynamics related to the creation and deletion of links over time. Let us recall that the fraction of created links at a given time is defined a the number of created links over the number of links *that could have been* created, that is according to the number of links that do not exist at preceding time. Similarly, the fraction of deleted links is defined at the number of deleted links over the number of existing links at preceding time.

The computation of those two quantities require then to memorize both the number of links that have been created (resp. deleted) at a given time but also the number of non existing links (resp. existing links) at previous time.

Exercise 19 Write a program that compute the fraction of created and deleted links at each time step and save the output in a file. When the fraction cannot be computed [1], the value will be -1.

1. For instance, if the graph is empty, one cannot compute the fraction of deleted links.