Significance of network metrics

Marta Arias, Ramon Ferrer-i-Cancho, Argimiro Arratia

Version 0.5
Complex and Social Networks (2018-2019)
Master in Innovation and Research in Informatics (MIRI)

1 Introduction

In this session, we are going to practice on determining the significance of network metrics using collections of global syntactic dependency trees from different languages. In those networks, the vertices are words and links indicate if two words have formed a syntactic dependency at least once in a syntactic dependency treebank [Ferrer-i Cancho et al., 2004] (Fig. 1). A syntactic dependency treebank is essentially a collection of sentences and their corresponding syntactic dependency trees. Here we focus for simplicity on the undirected versions of global syntactic dependency networks.

N is defined as the number of vertices of a network. The global clustering coefficient can be defined as [Newman, 2009]

$$C = \frac{\text{number of closed paths of length 2}}{\text{number of paths of length 2}}.$$

Alternatively, a mean local clustering was defined by Watts & Strogatz (WS) as [Watts and Strogatz, 1998]

$$C_{WS} = \frac{1}{N}\sum_{i=1}^{N} C_i, \tag{1}$$

where C_i is the local clustering of the i-th vertex, that is defined as

$$C_i = \frac{\text{number of pairs of different neighbors of } i \text{ that are connected}}{\text{number of pairs of different neighbours of } i}. \tag{2}$$

We adopt the convention that $C_i = 0$ if the degree of the i-th vertex does not exceed 2.

The mean closeness centrality is defined as [Newman, 2009]

$$\mathcal{C} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}_i, \tag{3}$$

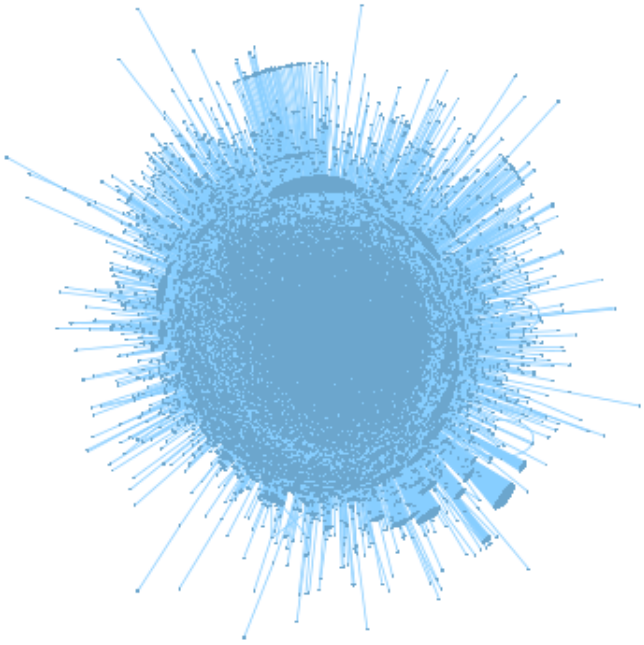

Figure 1: An English syntactic dependency network drawn with the Fruchterman-Reingold layout algorithm.

where $\mathcal{C}_i$ is the closeness centrality of vertex i, defined as

$$\mathcal{C}_i = \frac{1}{N-1} \sum_{j=1(i\neq j)}^{N} \frac{1}{d_{ij}},$$

being d_{ij} the geodesic distance between vertices i and j ($d_{ii} = 0$ and thus excluded from the summation; the purpose of that is that the $\mathcal{C}_i$'s and thus $\mathcal{C}$ have always finite values).

Through some procedure, two groups of student teams will be formed:

- The clustering group. Its teams will have to investigate if the clustering coefficient C_{WS} is significantly large.
- The closeness centrality group. Its teams will have to investigate if the closeness centrality $\mathcal{C}$ is significantly large.

Each team should work independently from other teams but is allowed to compare results with other teams of the group.

For this lab session, you do not need to use `R`. The reason is two-fold

- Time efficiency really matters for the exercises below.
- The algorithms may require some tuning or adaptations that are not *a priori* easy to apply using a standard graph library.

If you decide to not use `R`, we recommend `C` or `C++`.

2 Data preparation

The file `dependency_networks.tar.gz` included in this lab's package contains the description of the global syntactic dependency graphs from different languages. Each file consist of a header and a list of edges (the first row contains the number of vertices and the number of edges of the network; the other rows indicate the pairs of linked vertices). *Those networks may contain loops (a loop is an edge connecting a node with itself). Remove them before performing any analysis of the network properties.*

You have to produce a table with the format of Table 1. This table may be needed to interpret the results obtained in coming sections.

2.1 Test of significance

We want to determine if the value of a network metric x is significantly large with regard to a certain null hypothesis. x_{NH} is used to refer to the value of

Table 1: Summary of the properties of the degree sequences. N is the number of vertices of the network, E is the number of edges, $\langle k \rangle = 2E/N$ is the mean degree and $\delta = 2E/(N(N-1))$ is the network density of edges.

Language	N	E	$\langle k \rangle$	δ
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

x in a graph following a null hypothesis. We say that x is significantly large if $p(x_{NH} \geq x)$, the so-called p-value, is small enough, e.g. smaller than a significance level α. In this session two null hypotheses are considered:

- A binomial graph (Erdös-Rényi graph) with the same number of vertices and edges as the real network. This null model has no free parameter.
- A randomized graph with the same degree sequence of the original graph. The switching model is the randomization to use for this session. This null model has two parameters: the original network structure (the list of edges) and Q. The number of random switchings tried is QE, where E is the number of edges. The number of trials QE has to include cases where the random switching could not be performed. We advise you to tune Q according to the coupon collector's problem.

See further details on these null models in the corresponding theory lecture. **Our random or randomized networks cannot have loops or multiedges (pairs of vertices with more than one edge).**

Here you are going to estimate $p(x_{NH} \geq x)$ by means of the Monte Carlo procedure explained in the theoretical session, e.g., $p(x_{NH} \geq x) \approx f(x_{NH} \geq x)/T$, where T is the number of random graphs produced and $f(x_{NH} \geq x)$ is the number of those graphs where $x_{NH} \geq x$ (see the corresponding theory lecture for further details on the algorithm).

In order to implement calculations for the switching model successfully you have to answer the following questions (to be added to the report): given two edges $u \sim v$ and $s \sim t$, what are the switchings that

- preserve the degree sequence? (switchings not satisfying this property are not valid; performing them is a waste of time but they have to be counted to determine when QE switches, successful or not, have been reached)
- preserve the degree sequence but produce edges that are not allowed (loops, multiedges)?

Language	*Metric*	p-value (binomial)	p-value (switching)
...	...	...	...
...	...	...	...
...	...	...	...

Table 2:

Clue: consider the consequences of coincident vertices (edges sharing the same vertices, e.g., $u = s$).

Your are going to suffer the low speed of the computations. Thus,

- We recommend working on a single graph at the beginning. The smallest is the Basque one.
- We recommend using low values of T and Q at the beginning. The final value of Q should be of the order of 10 (at least). The final value of T must be greater than 20. The larger the value of T, the higher the accuracy of the estimated p-value. However, we admit that very large values are unfeasible for the whole collection of graphs.
- The p-value can be estimated or bounded doing some mathematical (analytical) work that goes beyond the scope of the this course. However, you can try. Those who succeed will be rewarded. Those who fail will not be penalized (in either case, the duty of a report cannot be skipped).

For the target network metric, you have to prepare a table with the value of the metric in the real network and the two estimated p-values, one for the null hypothesis of a binomial graph and another for the null hypothesis of the switching model. An example of the format is provided in Table 2.

3 Implementation

We strongly recommend that you think carefully about the data structures needed for storing the network information (e.g., an implementation based on the concept of an adjacency list is recommended).

We recommend a breadth-first search algorithm for computing the vertex-vertex geodesic distances (the d_{ij}'s).

3.1 Optimizations keeping results exact

Notice that in order to determine if $x_{NH} \geq x$, computing x_{NH} fully is not always necessary. Imagine that we have a way to bound x_{NH} below and above, x_{NH}^{min}

and x_{NH}^{max} respectively just by having explored a subset of the vertices and/or edges the network. Then if $x_{NH}^{min} \geq x$ then it can be concluded that $x_{NH} \geq x$. Similarly, if $x_{NH}^{max} < x$ then it can be concluded that $x_{NH} < x$.

Consider the case of $\mathcal{C}$. Imagine that the vertices are ordered arbitrarily and the we have actually computed $\mathcal{C}_i$ exactly for the M first nodes. Then, the definition of $\mathcal{C}$ in Eq. 3 can be rewritten as

$$\mathcal{C} = \frac{1}{N}\left(\sum_{i=1}^{M}\mathcal{C}_i + \sum_{i=M+1}^{N}\mathcal{C}_i\right). \tag{4}$$

Knowing that $0 \leq \mathcal{C}_i \leq 1$, lower and upper bounds to $\mathcal{C}$ can be inferred. On the one hand, $0 \leq \mathcal{C}_i$ yields

$$\begin{aligned}\mathcal{C} \geq \mathcal{C}^{min} &= \frac{1}{N}\left(\sum_{i=1}^{M}\mathcal{C}_i + \sum_{i=M+1}^{N} 0\right) \\ &= \frac{1}{N}\sum_{i=1}^{M}\mathcal{C}_i. \end{aligned} \tag{5}$$

On the one hand, $\mathcal{C}_i \leq 1$ yields

$$\begin{aligned}\mathcal{C} \leq \mathcal{C}^{max} &= \frac{1}{N}\left(\sum_{i=1}^{M}\mathcal{C}_i + \sum_{i=M+1}^{N} 1\right) \\ &= \frac{1}{N}\left(\sum_{i=1}^{M}\mathcal{C}_i + N - M\right) \\ &= \frac{1}{N}\sum_{i=1}^{M}\mathcal{C}_i + 1 - \frac{M}{N}. \end{aligned} \tag{6}$$

Thus, having computed $\mathcal{C}_i$ only for the M first vertices of the a network produced following a null hypothesis, if $\mathcal{C}_{NH}^{min} \geq \mathcal{C}$ then it can be concluded that $\mathcal{C}_{NH} \geq \mathcal{C}$; if $\mathcal{C}_{NH}^{max} < \mathcal{C}$ then it can be concluded that $\mathcal{C}_{NH} < \mathcal{C}$.

A clever ordering of the vertices might help to reduce the computation, for instance, by allowing you to detect that $C\mathcal{C}_{NH} < C\mathcal{C}$ earlier (with a smaller value of M). We suggest that you compare the speed of four orderings of vertices:

- Original ordering.
- Random ordering of vertices (by generating a uniformly random permutation of the vertices).
- Increasing order by degree.
- Decreasing order by degree.

The same ideas can be applied to the clustering coefficient C_{WS}. Notice that C_i is also a number between 0 and 1.

Calculations of C_i or $\mathcal{C}_i$ can be optimized when k_i, the degree of the corresponding vertex, is smaller than 2. Networks following a power-law-like degree distribution are expected to have many vertices with those degrees. On the one hand, recall that we have adopted the convention that $C_i = 0$ when $k_i < 2$. On the other hand, notice that

- if $k_i = 0$ then $d_{ij} = \infty$ for any vertex j and thus $\mathcal{C}_i = 0$.
- if $k_i = 1$, imagine that the i-th node is connected to the k-th vertex; then one has that $d_{ij} = d_{kj} + 1$. Thus knowing the minimum distance from k to other vertices one can derive the minimum distance from i to other vertices and vice versa. You will have to decide the direction of the inference that is more convenient.

3.2 Optimizations yielding approximate results

So far, we have assumed that x or x_{NH} have to be calculated exactly. It is possible to estimate x or x_{NH} faster but with some error through a Monte Carlo procedure. The key is that the error is small.

Imagine that the vertices of the network have been sorted producing a uniformly random permutation of the original vertices. Then, good estimates of the metrics can be obtained by just computing local metrics only for the M first vertices as

$$C_{WS} \approx \frac{1}{M} \sum_{i=1}^{M} C_i. \tag{7}$$

and

$$\mathcal{C} \approx \frac{1}{M} \sum_{i=1}^{M} \mathcal{C}_i. \tag{8}$$

Obviously, the estimation is perfect when $N = M$. Interestingly, a good estimation can be obtained even when $M \ll N$ (e.g., $100M/N = 10\%$ or even smaller could work).

4 Deliverables

You have to prepare a report including the following sections (in this order): introduction, results, discussion and methods. Results includes all the tables and some guiding text. Methods should include any relevant methods not explained in this guide (for instance, decisions that you had to made and might have an influence on the results), any clever decision to save computation time (e.g.

tighter bounds of C_{WS} or $\mathcal{C}$), the ordering of vertices used to bound the value of the metric, values of the parameters Q, T,...used, the ratio M/N used to estimate the metrics, etc. The discussion should include a summary of the results and your interpretation. For instance, you should discuss

- Under which null hypothesis the network metric is significantly large and under which it is not. Reason or speculate why this is so.
- The extent to which languages resemble or differ concerning the value of the metric or the results of the significance test.

You may need to refer to elementary network properties (Table 1) to interpret the results. The discussion section should also include some conclusions. The report should include the answer to questions above.

Important rule: Plagiarism will be prosecuted. Nevertheless, you are encouraged to ask the teacher as soon as possible if you think you do not understand what you are supposed to do, and also if you feel you are spending much more time than the rest of the group – sometimes a tiny error can be tricky to find and does not add much to your knowledge. Questions can be asked either in person or by email, and you will never be penalized by asking questions, no matter how stupid they look in retrospect.

To deliver: You must deliver the report explained above. The formats accepted for the report are, in principle, pdf, Word, OpenOffice, and Postscript. You also have to hand in the source code in `R` (or other languages) that you have used, including some minimal comments that can help the reader.

Procedure: Submit your work through the raco platform as a single zipped file.

Deadline: Work must be delivered within 2 weeks from the lab session you attend. Late deliveries risk being penalized or not accepted at all. If you anticipate problems with the deadline, please tell us as soon as possible.

5 Advanced exercices

Spend some time thinking about more powerful ways of bounding C_{WS} or $\mathcal{C}$ based on the decomposition of Eq. 4 or others. Reflect about the definition of C_i and $\mathcal{C}_i$. *A priori*, the most useful bounds as those that allow one to detect that $x_{NH}^{max} < x$ as one expects that the null hypothesis does not hold...but prior intuitions can fail.

Those are just some simple suggestions. You may find better approaches by your own.

References

[Ferrer-i Cancho et al., 2004] Ferrer-i Cancho, R., Solé, R. V., and Köhler, R. (2004). Patterns in syntactic dependency networks. *Physical Review E*, 69(5):051915.

[Newman, 2009] Newman, M. (2009). *Networks: an introduction*. Oxford University Press.

[Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440.