Lab 3: Significance of network metrics

Rodrigo Arias Mallo

November 2, 2017

1 The metric

For selecting the metric, I set the seed to the last 5 digits of my DNI, as recently seen in some hash generation[1], and ran the following command in python:

```python
from random import *

seed(64718)
metrics = ["clustering coefficient", "closeness centrality"]
r = randint(0, 1)

print(metrics[r])
```

Which produced the following output

```
% python metric.py
clustering coefficient
```

So I decided to use clustering coefficient C_{WS} as a metric.

2 Introduction

2.1 Cleaning the data

The datasets have been uncompressed in `data/`, and then processed by the script `prepare-data.sh`, which removes the header with the number of nodes and edges, and outputs the remaining edge list in the same file with a new extension `.edges`. The properties of the graphs can be seen at the table 1.

After some research I found the BiRewire R package [1] written by A. Gobbi, a fast implementation of the switchin model [2] [3].

3 Proposition

Based on some properties of a graph $\hat{G}$, we want to create a sequence of random graphs $\langle G \rangle$ by using two different methods, and then test if they maintain a set of measures that we found on $\hat{G}$.

The first model, the Erdös–Rényi graph takes as input the number of vertex $|V|$ and edges $|E|$ of $\hat{G}$, and builds a new random graph with the same number

[1] `http://klondike.es/klog/2017/09/25/descifrando-las-bases-de-datos-del-referendum-catalan/`

Language	N	E	$\langle k \rangle$	δ
Arabic	21531	68742	6.385	2.966×10^{-4}
Basque	12207	25541	4.185	3.428×10^{-4}
Catalan	36865	197075	1.069×10^{1}	2.900×10^{-4}
Chinese	40298	180925	8.979	2.228×10^{-4}
Czech	69303	257254	7.424	1.071×10^{-4}
English	29634	193078	1.303×10^{1}	4.397×10^{-4}
Greek	13283	43961	6.619	4.984×10^{-4}
Hungarian	36126	106681	5.906	1.635×10^{-4}
Italian	14726	55954	7.599	5.161×10^{-4}
Turkish	20409	45625	4.471	2.191×10^{-4}

Table 1: Properties of the graphs after preproccesing.

of vertex and edges. The clustering coefficient is then computed for each graph in the sequence $\langle G \rangle$ as $\langle X \rangle$.

We can consider the measurement X as a random variable, with mean $E[X]$ and variance $VAR[X]$. By computing T elements in the sequence, the sample mean $\overline{X}$ is an unbiased estimator of $E[X]$, and by the central limit theorem, the sample $\overline{X}$ is distributed

3.1 Erdös–Rényi model

The ER model is implemented in `python` using the `networkx` package. The `gnm_random_graph` creates a ER graph with parameters $|V|$ and $|E|$. An average of $T = 25$ graphs is performed. The measure is taken by calling `average_clustering`. We see that none of the generated graphs contain a value greater that the original one. We can conclude that, even the ER model keeps the number of nodes and edges, the clustering coefficient is smaller.

Language	x	$\overline{x}_{ER}$	$p(x_{ER} \geq x)$
Arabic	1.885×10^{-1}	2.958×10^{-4}	0.000
Basque	4.671×10^{-2}	2.971×10^{-4}	0.000
Catalan	2.211×10^{-1}	2.916×10^{-4}	0.000
Chinese	1.708×10^{-1}	2.310×10^{-4}	0.000
Czech	1.217×10^{-1}	1.124×10^{-4}	0.000
English	2.353×10^{-1}	4.416×10^{-4}	0.000
Greek	1.338×10^{-1}	4.800×10^{-4}	0.000
Hungarian	5.085×10^{-2}	1.549×10^{-4}	0.000
Italian	1.437×10^{-1}	4.868×10^{-4}	0.000
Turkish	2.236×10^{-1}	2.317×10^{-4}	0.000

Table 2: The measures of ER model.

Language	x	$\overline{x}_S$	$p(x_S \geq x)$
Arabic	0.188	0.187	0.200
Basque	0.047	0.053	1.000
Catalan	0.221	0.145	0.000
Chinese	0.171	0.093	0.000
Czech	0.122	0.070	0.000
English	0.235	0.240	1.000
Greek	0.134	0.147	1.000
Hungarian	0.051	0.072	1.000
Italian	0.144	0.198	1.000
Turkish	0.224	0.247	1.000

Table 3: The measures of the switching model.

4 Results

5 Discussion

6 Methods

References

[1] A. Gobbi, F. Iorio, D. Albanese, G Jurman, and J. Saez-Rodriguez. *BiRewire: High-performing routines for the randomization of a bipartite graph (or a binary event matrix), undirected and directed signed graph preserving degree distribution (or marginal totals)*, 2017. R package version 3.8.1.

[2] A. Gobbi, F. Iorio, K.J. Dawson, D.C. Wedge, D. Tamborero, L. Alexandrov, N. Lopez-Bigas, M.J. Garnett, G Jurman, and J. Saez-Rodriguez. Fast randomization of large genomic datasets while preserving alteration counts. *BMC Bioinformatics*, 30(17):617–623, 2014.

[3] F. Iorio, M. Bernardo-Faura, A. Gobbi, T. Cokelaer, G Jurman, and J. Saez-Rodriguez. Efficient randomization of biologicalnetworks while preserving functionalcharacterization of individual nodes. *BMC Bioinformatics*, 17(1):617–623, 542.