Lab 6 - CSN: Network dynamics

Pierre-Antoine Porte
porte.pierreantoine@gmail.com

Rodrigo Arias Mallo
rodarima@gmail.com

December 12, 2017

Introduction

In this session, we had to model 3 different models following the dynamical principles of the Barabasi-Albert model. Those principles are: vertex growth and preferential attachment. The different models we needed to implements were the following:

- Barabasi-Albert with the dynamical principles
- Barabasi-Albert with random attachment instead of preferential attachment (only one dynamical principle).
- Barabasi-Albert with vertex growth suppressed (only one dynamical principle).

Those models were simulated and the data kept in files so we could analyze mathematical properties of those models.

In this report we will show our results regarding those models and their analysis. We will discuss those results and explain them. Then we will explain how we implemented the model simulations and what we used to analyze it in the Methods section.

Results

	1	2	3
1	0.000	0.000	27.383
2	70.469	85.777	0.000
3	125.185	123.279	∞
4	30.885	110.386	∞
5	138.677	129.436	∞

Table 1: AIC measures for the degree distribution.

Discussion

As explained in section Methods, we used m_0 and n_0 always with the same values (which can be however different for each model). We never compared the models with different m_0 or n_0 for the same model. It could have indicate us if the model behave differently given different graph in input. We could have went further and do this but we preferred to focus on the analysis of our different models.

Methods

For the default *Barabasi-Albert* and model with random attachment the initial graph was an empty graph with only one vertex.

For the model without vertex growth, we used an unconnected graph with t_max vertices. Because we have no vertex growth, the vertices are not increasing and $n_0 = n_tmax$.

For the three models we used $m_0 = 0$ to used "*clean*" and "*empty*" graphs.

We measured the growth of the vertex degree over time and the degree distribution for each model. The vertex degree was measured over the time for $t = 1, 10, 100, 1000, 10000$ successively.

We used python for generating the models, to store the results and to analyze the data. For each BA model M a folder in `data/model`M`/` contains all the results produced from this model. Inside, the degree sequence is stored in the file `dseq.txt`, the degree distribution in `dd.txt` and for each T in the arrival time, we produced `dt_`T`.txt` tracing the degree of the vertex arriving at time T.