Lab 6: Network Dynamics

Paco Jurado Moreno and Miguel Alcón Doganoc

December 9, 2019

Introduction

The goal of this lab session is to simulate different network growth models and analyze their properties from a statistical perspective. The main objectives are: achieve a better understanding of the dynamical principles behind the Barabási-Albert model, improve our simulation skills, and apply curve-fitting methods of previous sessions.

We borrow the notation from the theoretical sessions: t indicates the timestep ($t \geq 0$) and t_i refers to the time at which the $i-th$ vertex appeared .In this session, we also use t_{max} to indicate the last timestep of the simulation. The model has two parameters:

- n_0: the initial number of vertices.
- m_0: the initial number of edges of every new vertex.

Regarding the simulation part, we want to analyze the mathematical properties of the Barabási-Albert model and two modified versions: one where preferential attachment is replaced by random attachment, and another where vertex growth is suppressed. We focus on the study of the growth of vertex degrees over time (time series) and their degree distribution.

Results

Discussion

Methods

Simulation of the Barabási-Albert model

Language and random genenerator

As it is recommended in the guide, we developed the simulation of the Barabási-Albert model in C++. We also used the `random` library from the Standard Template Library of C++ to generate random decisions. More specifically, the `uniform_int_distribution` function, together with the *Mersenne Twister 19937* generator.

Time series

Time series are collected at times 1, 10, 100 and 1000.

Initial configuration of the network

Since there is no indication of how to create the network at time 0, we decide to generate a cycle graph randomly. Basically, given n_0, the algorithm orders the n_0 nodes randomly. Then, it connects the first with the second, the second with the third, etc. until all of them are connected. Finally, it connects the last one with the first one. With this, we start with a fully connected network, whose all vertices have the same degree (2). The latter implies that all nodes have the same number of occurrences in the stubs vector (when it is used), which means they have the same probability of being chosen.