Network Science

Lab #2
Network models

Timetable

- Lab 1 – Fri Oct 12
  Scale free properties
- Lab 2 – Fri Oct 19
  Albert-Baràbasi model
- Lab 3 – Fri Oct 26
  Assortativity
- Lab 4 – Fri Nov 16
  Ranking
- Lab 5 – Fri Nov 23
  Community detection – Spectral
- Lab 6 – Fri Nov 30
  Community detection – PageRank-Nibble
- Lab 7 – Fri Dec 7
  Gephi

MATLAB Licence

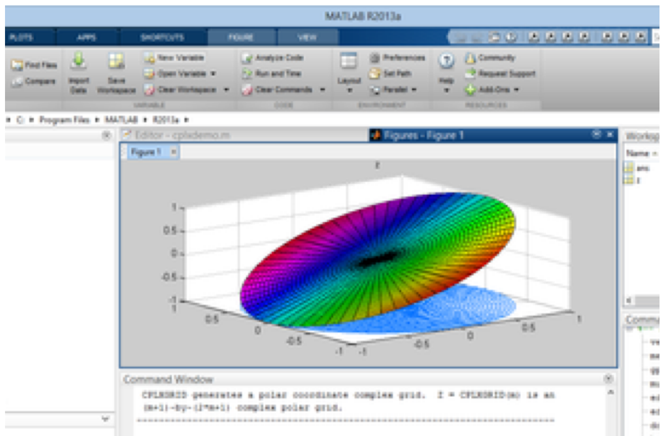
MATLAB = MATrix LABoratory by MathWorks

MATLAB *"is a numerical computer environment which allows matrix manipulations, plotting of functions and data, implementation of algorithms"* [wiki]

Total Academic Headcount
Campus & Student

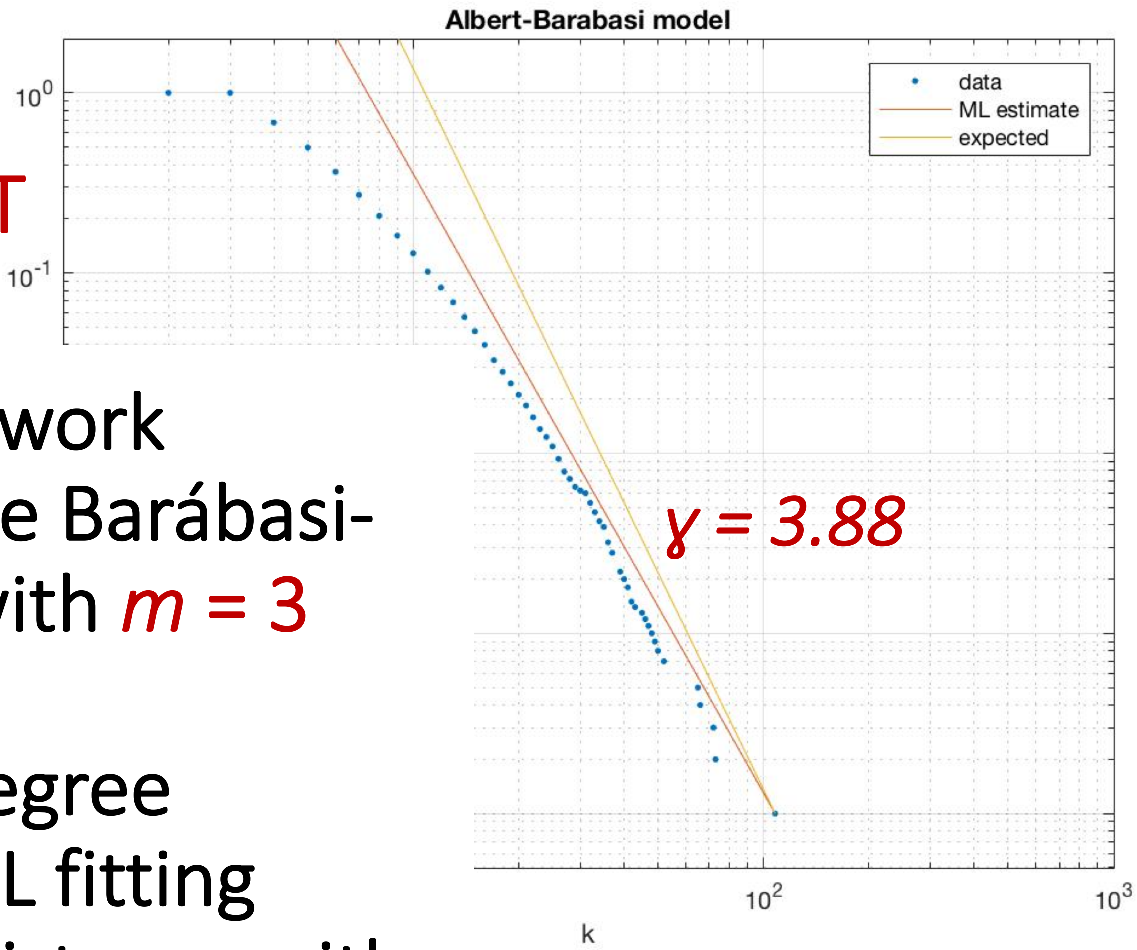
You can freely install MATLAB in your laptop.

https://www.csia.unipd.it/servizi/servizi-utenti-istituzionali/contratti-software-e-licenze/matlab



Lab 2 – Assignment 1

BARABASI ALBERT



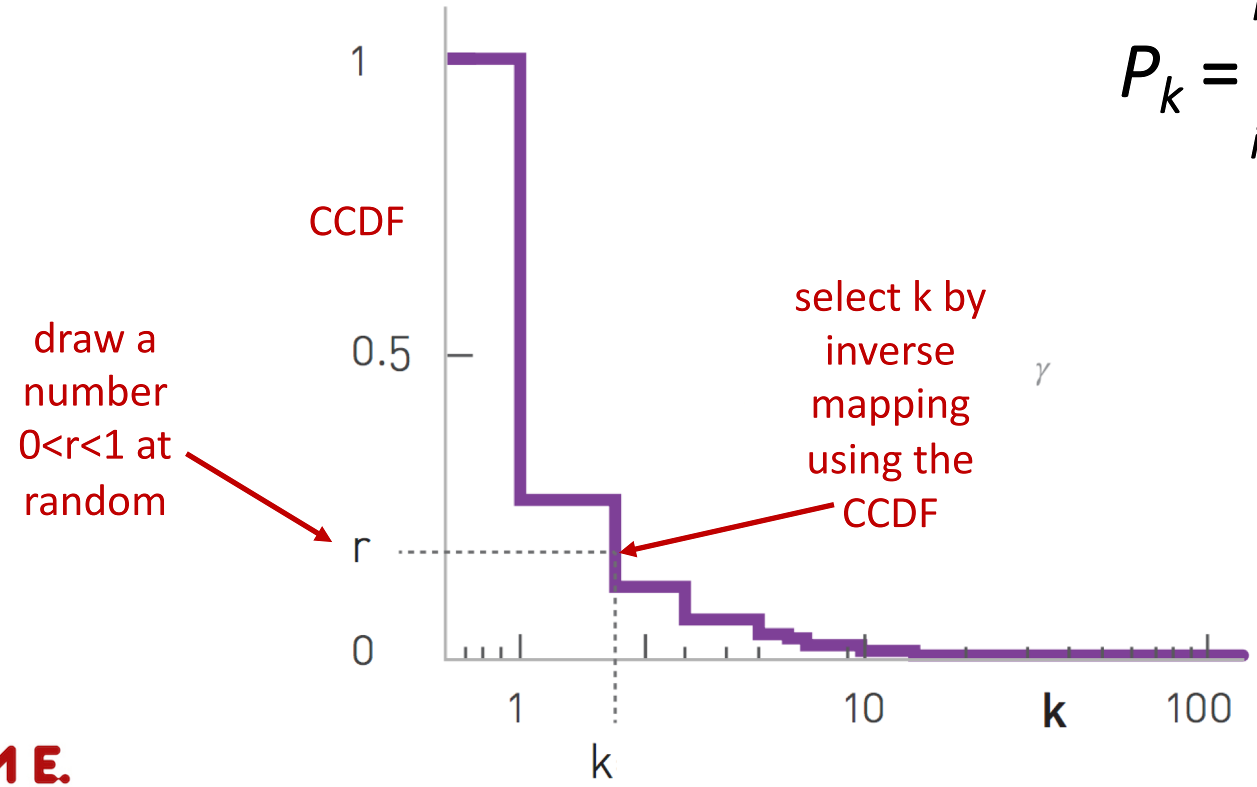
1. Generate a network according to the Barábasi-Albert model with $m = 3$ and $A = 3$
2. Estimate the degree exponent by ML fitting
3. Check the consistency with the expected $\gamma = 3 + A/m$

Lab 1 – MatLab hints

1. Sum: sums a (sparse) matrix by columns or rows
2. Unique: finds the unique elements of a vector
   unique([1 2 3 2 1]) = [1 2 3]
3. Cumsum: cumulative sum
   cumsum([1 2 3 2 1]) = [1 3 6 8 9]
4. Mean: computes the average
5. Histc: counts occurrences in a vector
   histc([0.5,0.9,1.3],[0 1 2]) = [2 1 0]
   histc([-1,0,1,2,3],[0 1 2]) = [1 1 1]
6. Loglog: logarithmic plot

Lab 2 - RVs with a given distribution

- How to generate degrees with given distribution p_k?
- Use the inverse CCDF method



$$P_k = \sum_{i=k}^{k_{max}} p_i$$

The method [edit]

The problem that the inverse transform sampling method solves is as follows:

- Let X be a random variable whose distribution can be described by the cumulative distribution function F_X.
- We want to generate values of X which are distributed according to this distribution.

The inverse transform sampling method works as follows:

1. Generate a random number u from the standard uniform distribution in the interval $[0,1]$, e.g. from $U \sim \mathrm{Unif}[0,1]$.
2. Find the inverse of the desired CDF, e.g. $F_X^{-1}(x)$.
3. Compute $X = F_X^{-1}(u)$. The computed random variable X has distribution $F_X(x)$.

Expressed differently, given a continuous uniform variable U in $[0,1]$ and an invertible cumulative distribution function F_X, the random variable $X = F_X^{-1}(U)$ has distribution F_X (or, X is distributed F_X).

A treatment of such inverse functions as objects satisfying differential equations can be given.[4] Some such differential equations admit explicit power series solutions, despite their non-linearity.[citation needed]

Lab 2 – Further MatLab hints

1. Rand: generates random numbers uniformly distributed in (0,1)

2. To generate *m* random values according to a PDF contained in vector *p* do

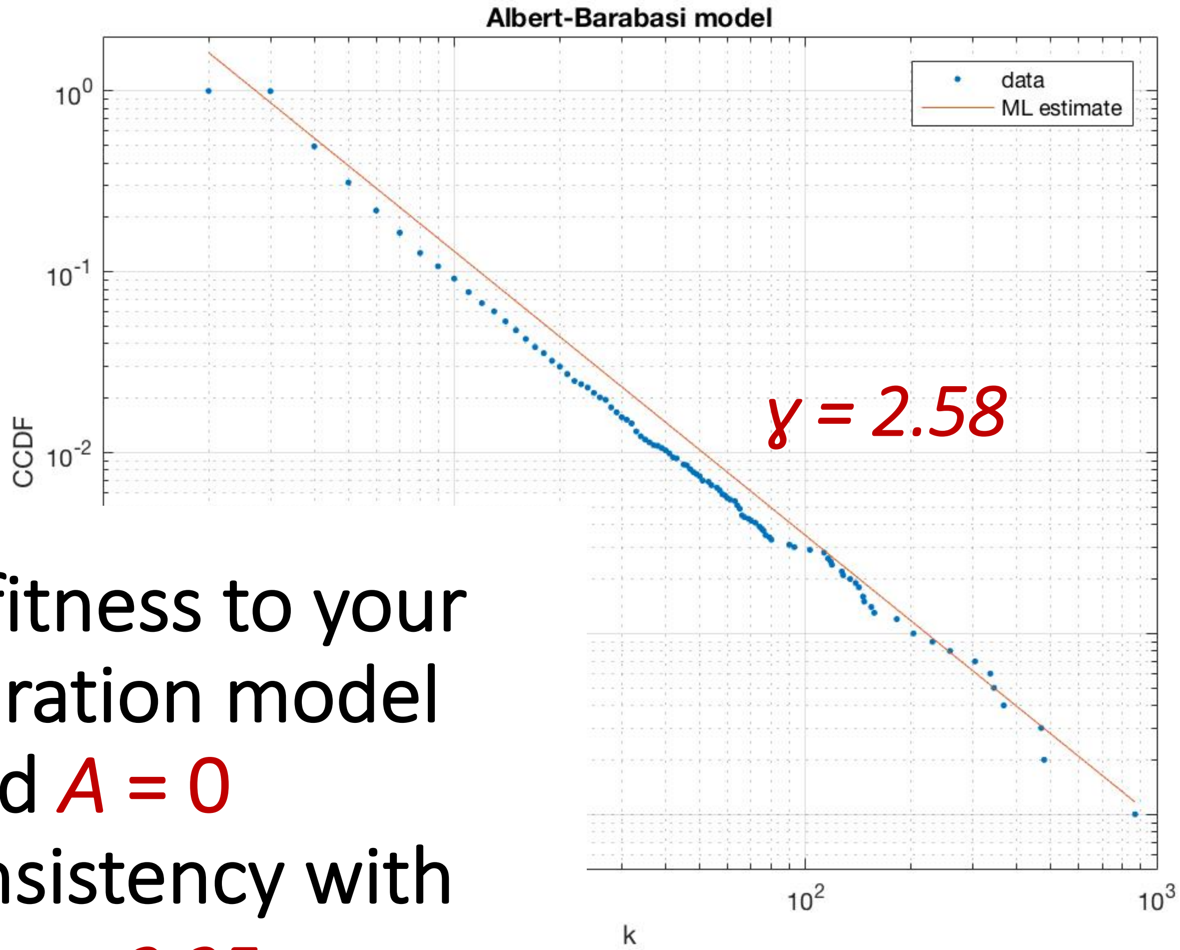
   ```
   x = rand(1,m);             m random values
   edges = cumsum([0, p]);    edges of the CDF
   N = histc(x,edges);        counts the number of occurrences
                              per interval (discard the last value!)
   ```

3. Keep matrix A sparse !!!

MiME.

Lab 2 – Assignment 2

FITNESS



1. Add uniform fitness to your network generation model with $m = 3$ and $A = 0$
2. Check the consistency with the expected $\gamma = 2.25$
3. What if fitness is exponential?

Lab 2 – Further MatLab hints

1. Rand: generates random numbers uniformly distributed in [0,1]

2. -log(rand): generates random numbers with an exponential distribution