**EE 232E**

**Graphs and Network Flows**

Homework 1 Report

Pingyuan Yue: 504737715

Ke Xu: 604761427

Yuanyi Ding: 404773978

4/17/2017

Table of Contents

[**1. Create Random Networks 3**](#_Toc480208159)

[**2. Dataset and Problem Statement 4**](#_Toc480208160)

[**3. Creates a Random Graph by Simulating Its Evolution 4**](#_Toc480208161)

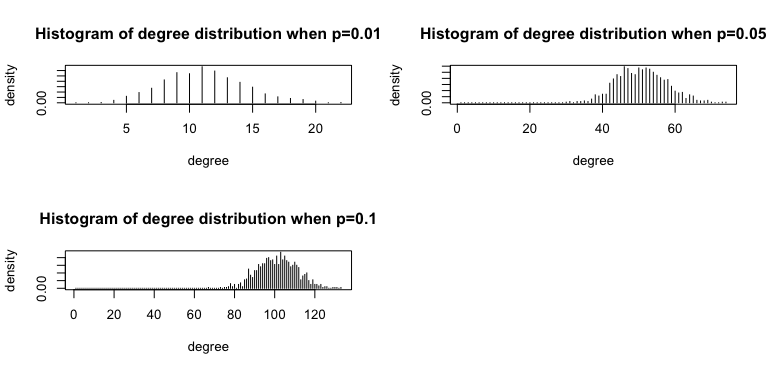
[**4. Forest Fire Model 5**](#_Toc480208162)

[**5. Conclusion 6**](#_Toc480208163)

# Create Random Networks

**Part (a):**

We are asked to create three undirected random networks with 1000 nodes, and the probability p for connecting two arbitrary vertices 0.01, 0.05 and 0.1 respectively. Below are three figures of degree distribution:



Figrue1.1: degree distribution for different value of p

We observed from the graphs above that the networks trend to have high degree if the probability p for connecting two arbitrary vertices is high.

**Part (b):**

Due to fact that our graphs are randomly generated, we decide to repeat the process for 100 times and compute the average for connectivity and diameter. The results are shown as following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | P = 0.01 | P = 0.05 | P = 0.1 |
| Connectivity | 0.99 | 1 | 1 |
| Diameters | 5.34 | 3 | 3 |

Table 1.1: connectivity and diameters for different value of p

**Part (c):**

We are asked to find out a value of , so that any networks we generated with a p that is smaller than are disconnected, and any networks generated with a p that is larger than are connected. Considering the results in part b, we decide to perform a binary search between 0 and 0.01 to find the . The algorithm stops whenever the graph we generate is disconnected with possibility of (p - ε) and is connected with possibility of (p + ε). That value of p is the we try to find. The we find is 0.00728.

**Part (d):**

We are asked to derive the value of analytically. We assume the with certain value of p, we expect the find one single isolated node. The expectation of finding this node can be represented as:

1 = , where N = 1000

p =

The p we calculate from the above equation is equal to 0.00688, which is close to what we get from part c.

# Dataset and Problem Statement

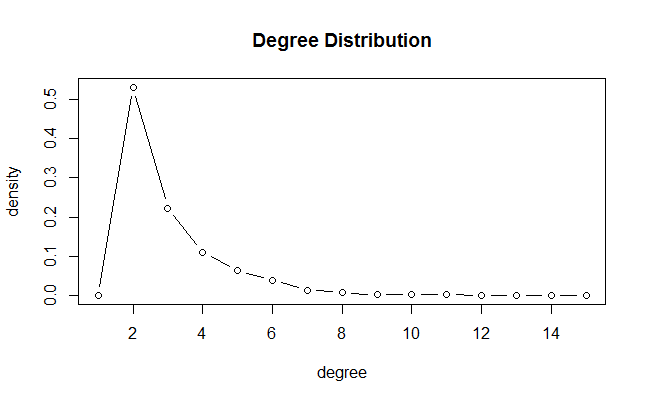
# Creates a Random Graph by Simulating Its Evolution

1. By definition, a preferential attachment process is any of a class of processes in which some quantity, typically some form of wealth or credit, is distributed among a number of individuals or objects according to how much they already have, so that those who are already wealthy receive more than those who are not.

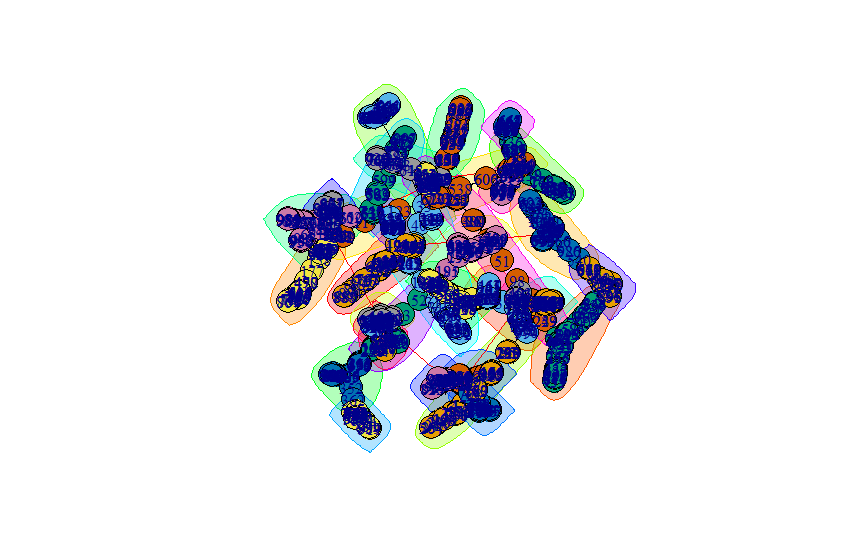
In this problem, we created a random graph with 1000 nodes by simulating its evolution. A new vertex is added and creates a number of links to old vertices and the probability that an old vertex is cited depends on its in-degree. We can simply simulate the evolution of the graph by calling the sample\_pa\_age function in igraph library.

g <- sample\_pa\_age(1000, pa.exp=1, aging.exp=-1, aging.bin=1000, directed=FALSE)

The degree distribution of the graph is shown as follows:

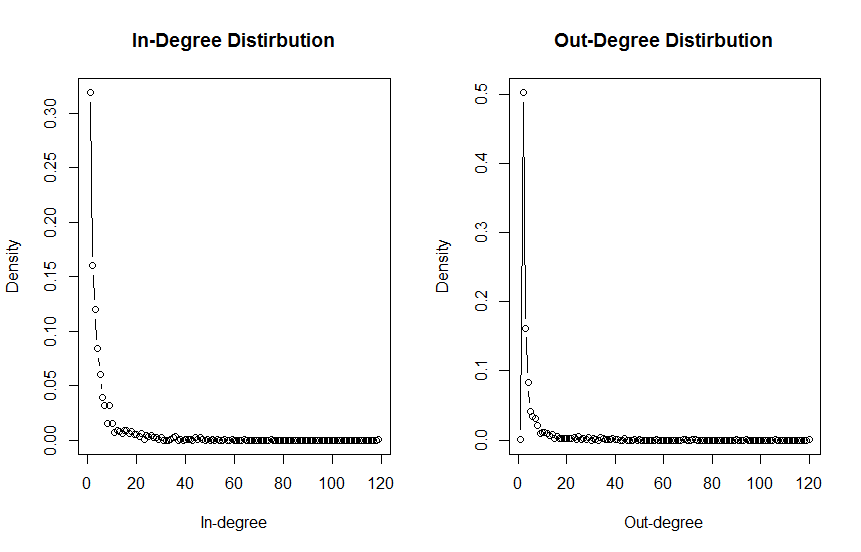


1. Use fast greedy method to find the community structure. We can find that the number of communities of this graph is around 32. The modularity is 0.93597. The structure of the graph is shown below:

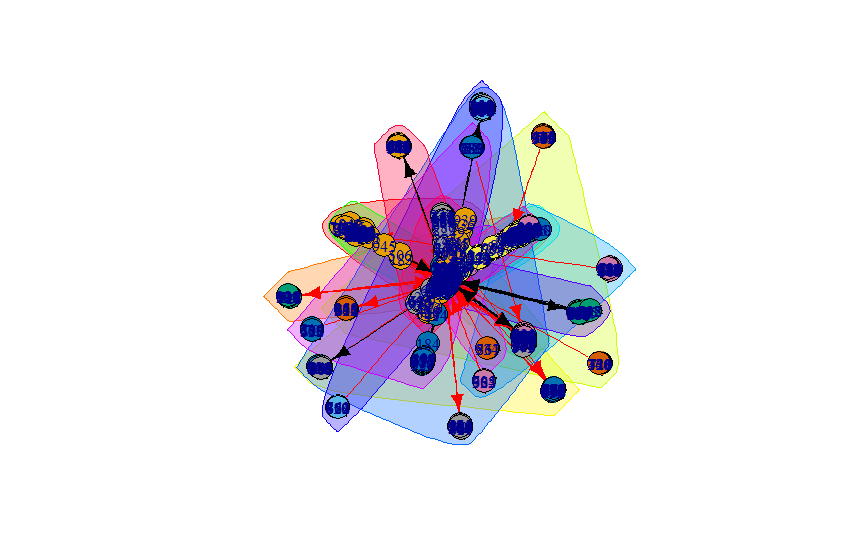


# Forest Fire Model

1. In this problem, we generate a forest fire model by using forest.fire.game function. The forward burning probability is set to be 0.37, and the backward burning probability is set to be 0.32/0.37 = 0.86. Both in-degree distribution and out-degree distribution of the graph are shown below. It could be concluded that both degree decrease quite fast.



1. We generate the graph for 50 times and calculate the average diameter which is around: 10.56.
2. In this problem, the structure of the graph is calculated using spinglass method, since fast greedy method is only applicable to undirected graph. The number of communities is around 24 and the modularity of the graph is calculated to be 0.47009. The structure of the graph is shown below:



# Conclusion

In this assignment, we learn the basic knowledge of several graph theories and models and get familiar with igraph library and R language programming. The main challenges in this assignment is to figure out the meaning of different models (e.g. fat-tailed distribution, preferential attachment, forest fire) find the correct method to generate different models.