Lab L5: Dynamic processes on graphs

Shaoyong Guo
Politecnico di Torino
s296966
s296966@studenti.polito.it

I. Introduction

The purpose of this simulation is to simulate dynamical processes on graphs. In fact, there are two closely related variants of the ER random graph model, G(n,P) and G(n,M). In my simulation, the G(n,P) model is chosen to simulate the process of graph theory. Basically, the degree distribution of any particular vertex in a graph G(n, p) is a binomial distribution. I will test this theory through simulations.

II. The simulation model

A. Stochastic Elements

Theoretically, in a graph G(n,P) model with n nodes, each vertex generates an edge with a certain probability, which is connected to other vertices in the graph. At the beginning of simulation, it is necessary to set the initial condition.

The number of nodes. In the simulation, I need to set the number of nodes in advance. This is the basis of the simulation, which will generate a different number of edges for each vertex with fixed probability. But this does not mean that every vertex will have an edge. In the simulation, all information about the nodes (nodes, edges) is recorded in the event list. In my simulation, the number of nodes = 500.

The probability for generating edges. The probability determines whether a vertex will produce an edge. As p increases from 0 to 1, the model increasingly favors graphs that contain more edges and increasingly disfavors graphs that contain fewer edges. I use a uniform probability for the simulation.

The Poisson Distribution. In the simulation, I set the random Poisson distribution probability to help determine if vertices can generate edges. When the random Poisson probability is above the random uniform probability, the graph generates edges between different vertices

B. Input Parameters

The list of Events. The list of events generated by the simulation consists of row i and column j. At the beginning, all nodes in row i and j are 0. At the beginning, all nodes in rows i and j are 0. For random vertices, the system generates a random Poisson probability and a random uniform probability. If the random Poisson probability is higher than the random uniform probability, the vertex will be converted to 1.

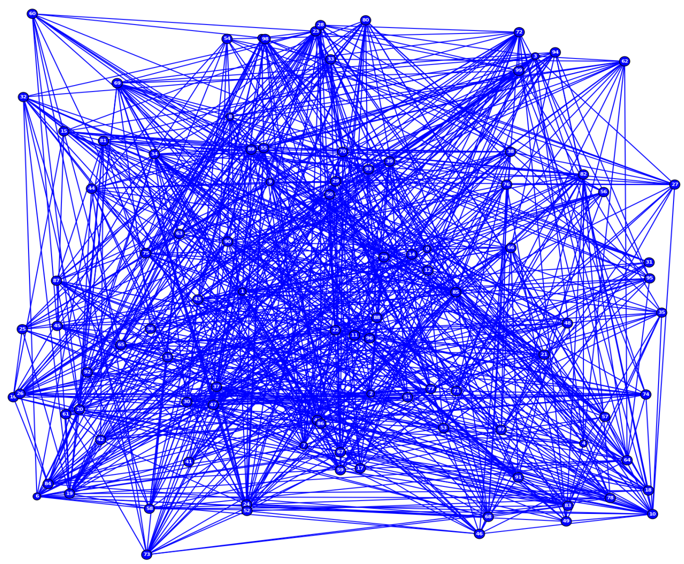

Fig. 1. The Graph of Nodes and Edges

Position Dictionary. I created a location dictionary with a key i and a pair of random values. If the value of an event is 1, the index of the event is chosen as the key of the location dictionary. In Figure 1, all nodes and edges are labeled.

C. Algorithm

The distribution of the degree of vertex. Theoretically, the degree of all vertex is will be follow a binomial distribution:

$$P(deg(v) = k) = \binom{(n-1)}{k} p^{k} * (1-P)^{(n-k-1)}$$

where n is the total number of vertices in the graph. If n tends to be infinite, the distribution will be Poisson.

D. Output Metrics and Results

The distribution of degree In the Fig.2, the empirical distributions of all degrees are represented by histograms, which fit a true binomial distribution.

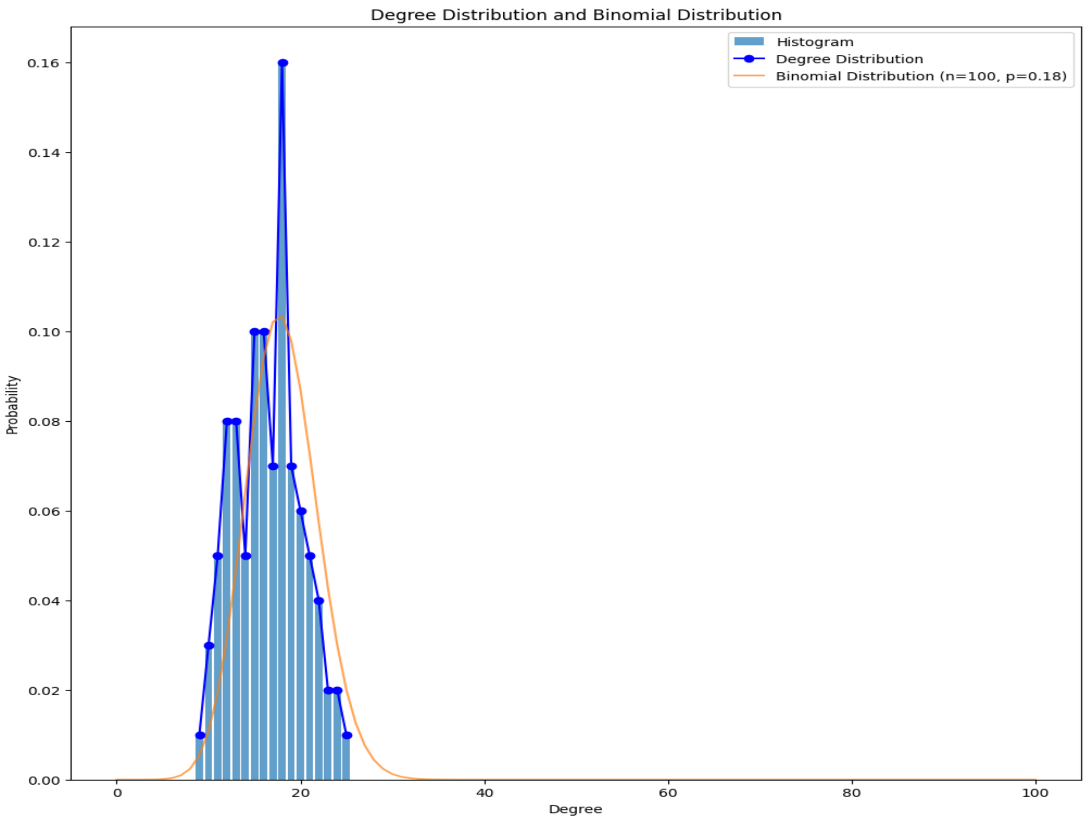

Fig. 2. The Distribution of Degrees

The quantile–quantile test of degree. In the Fig.3, all of the degree distribution data fit well into the expected binomial theoretical distribution. **The results of Chi-square Test.** In my Chi-square hypothesis testing, the p-value = 1 and chi Stat = 0.0.

The P value = 1 indicates that there is no statistical evidence against the null hypothesis. And chi Stat = 0.0 indicates that there is no difference between the empirical and expected values. This result demonstrates that the degrees of each vertex being compared are independent and there is no correlation between them.

III. Conclusion

In my simulations, I employed uniform and Poisson probabilities to generate random edges for the vertices. Then, quantile-quantile tests and chi-square tests were adopted to assess whether the empirical degrees of the vertices conform to the theoretical empirical degrees of the vertices.

The final results showed that all results were as expected.

The requirement is n = 100k and the p = $10^{-}4$ in our experiment. However, I didn't comply with this condition because I couldn't make the simulation more efficient. It would have taken a lot of time to use n = 100k. I will solve all the problems in the next simulation.

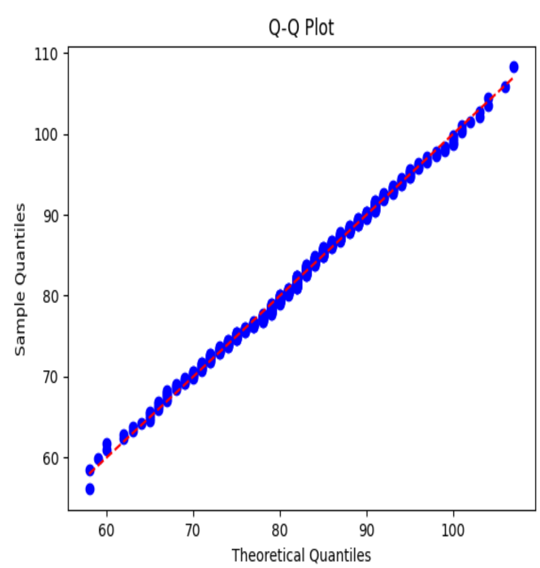

Fig. 3. The Quantile-Quantile of Empirical and Theoretical Degree