Parallel and Distributed Systems - Assignment 4

Authors:
Giannouris Polydoros AEM: 9746
Grigorakis Efthymios AEM: 9694

1 Introduction

Networks come in many forms, but there is no denying they are all around us. From computer science to sociology and biology, more and more scientific fields use graphs to represent the inner workings that define large structures. Naturally occurring networks have been found to have many properties including, but not limited to, the small world, the clustering and the heavy tailed degree distribution property. Several random models have been proposed that address one or more of these properties, three of which will be analyzed with respect to their community structures in this assignment. Specifically we will study the scale-free, the small-world and the Erdős–Rényi models. You can find our code on Github.

2 Evaluating a partition

In order to group a graph's nodes into communities we must use a function that defines how good a certain partition is. The modularity function of a community achieves this by comparing the number of edges inside the community, as opposed to those connecting it with other ones. However discerning sub-communities inside larger ones is deemed impossible. Instead a better solution is the Constant Potts Model, defined as

$$H = \sum [e_c - \gamma * \binom{n_c}{2}]$$

where n_c is the number of nodes for the cummunity that CPM is being computed, e_c is the actual number of edges in community c and γ is the minimum density of the community. This introduces a second criterion of keeping community sizes small, in addition to maximising edges within the community and minimising connections with other communities, effectively solving the aforementioned issue.

3 Louvain Algorithm

The Louvain Algorithm aims to create an ideal partition using one of the methods above. Initially each node is assigned to a community. After that, pairs of nodes are swapped between communities in an attempt to maximise modularity or CPM. Once maximized, each community is condensed into a single node containing all edges of the initial community. Repeating these steps until no further improvement can be made, leads to the clustering configuration.

4 Leiden Algorithm

In this assignment we will use the Leiden algorithm, an improved version of the Louvain algorithm. Both use either the modularity or the CPM. However the Leiden algorithm instead of merging the node with the community that yields the best partition, randomly selects one of all possible communities that offers a better partition. Although this may produce a locally worse partition, it enables us to look for more partitions, some of which will be better than the ones produce by the Louvain algorithm. Additionally several methods aiming to speed up the basic algorithm are implemented, although delving into those details is not one of the goals of this assignment. After using the Leiden algorithm on the random graphs we will display the adjacency matrix based on the produced clustering configuration, accompanied with its interpretation. Alongside each matrix is the corresponding histogram of the sizes of its communities.
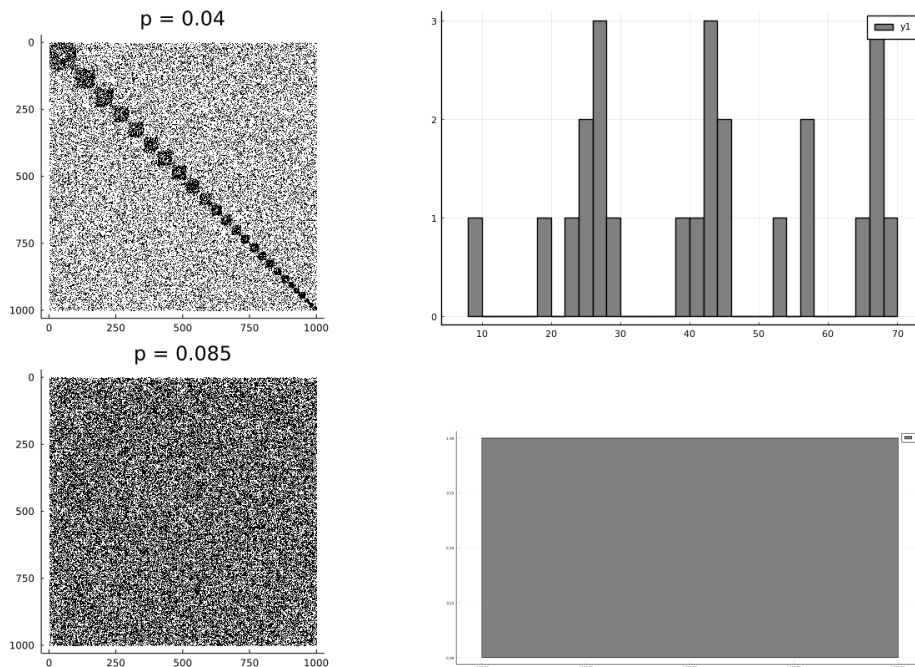
5 The Erdős–Rényi model

The Erdős–Rényi model is the simplest of the random graphs studied. Using the notation **G(n,p)**, the graph is produced by connecting any two of the total **n** nodes with a probability of **p**. As p approaches 1 the graph more closely resembles a full graph, while it has been shown that

$$\frac{\ln n}{n}$$

is the threshold of connectedness.
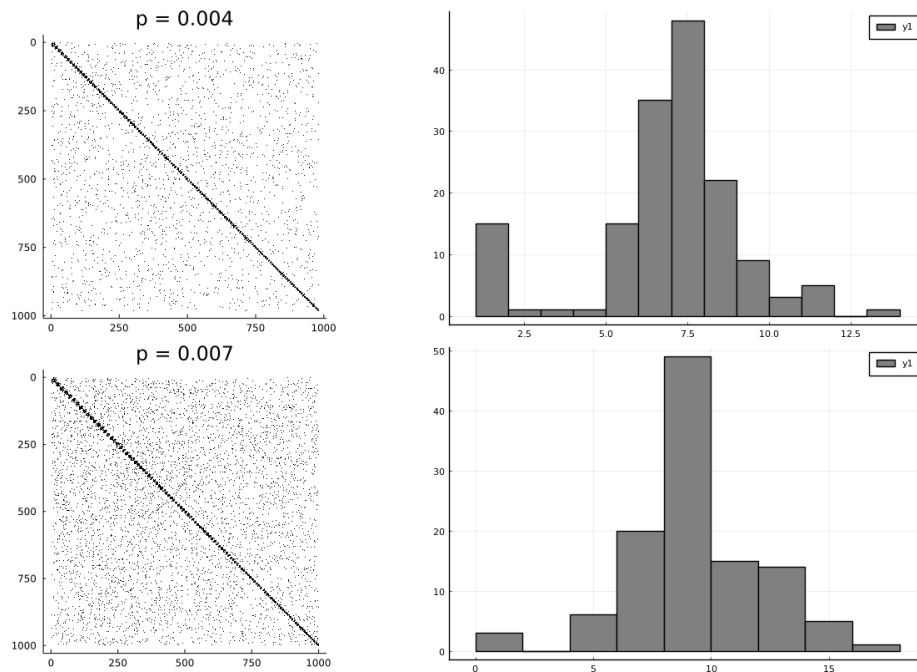
Adjacency matrix view for various values of p and n = 1000.

Notice that as p increases, the size of the communities also increases, until the graphs gets dense to the point of forming one huge community.

6 The small-world model

The small-world model aims to introduce short average path lengths and high clustering. To achieve this Watts and Strogatz combined regular ring lattices with the Erdős–Rényi model. To produce the random graph we have to start from a ring lattice of **n** nodes each connected with its **k/2** nearest neighbours on each side. Afterwards we rewire every edge (i,j), on one side of each node, to a random j' with a probability of **b**.

Adjacency matrix view for various values of k, b = 0 and n = 1000.
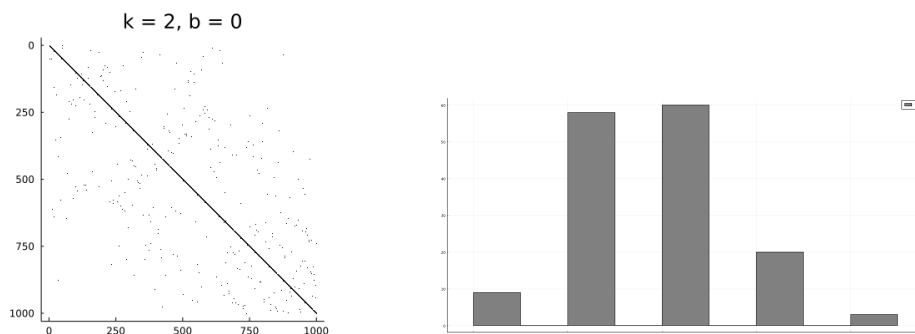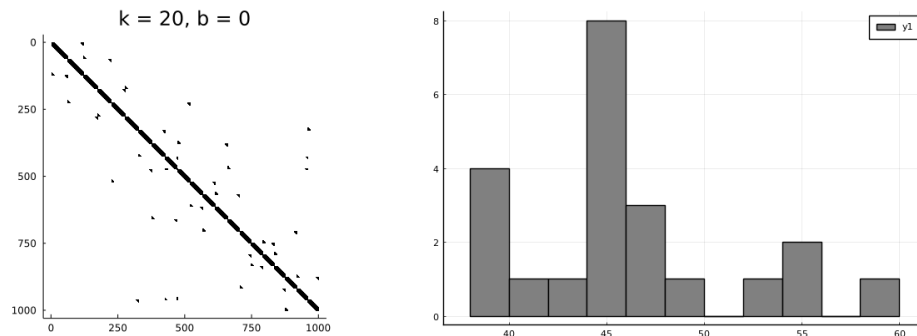
With a **b** of zero, no edges are rewired, resulting in a ring lattice that, as expected, has an inversely proportional number of communities to **k**.
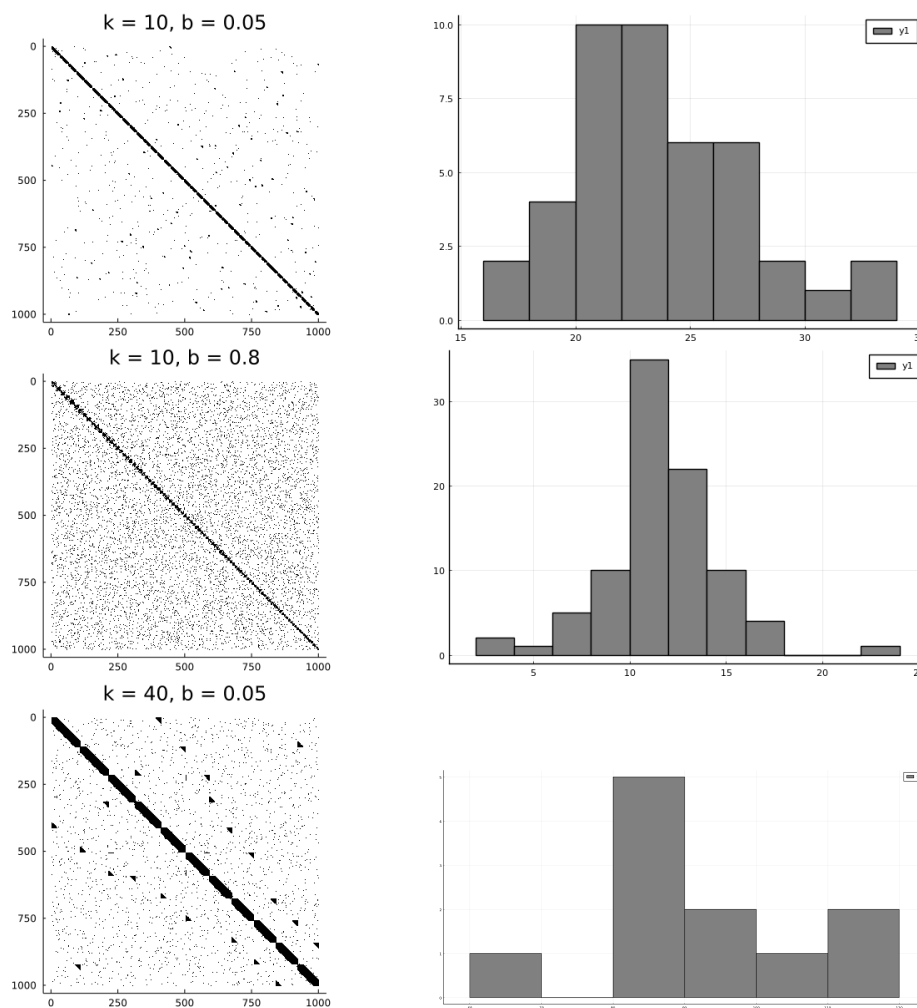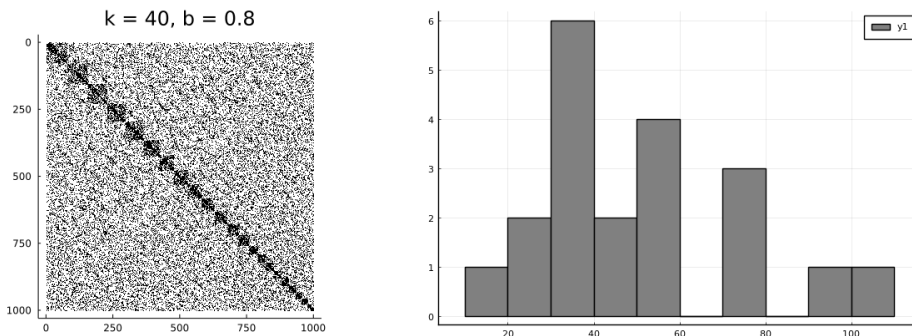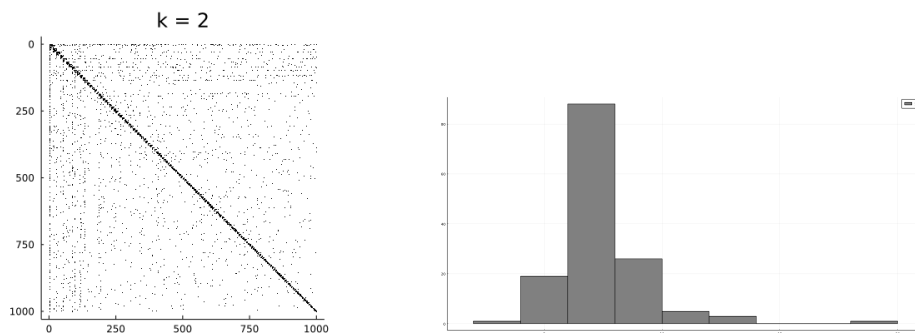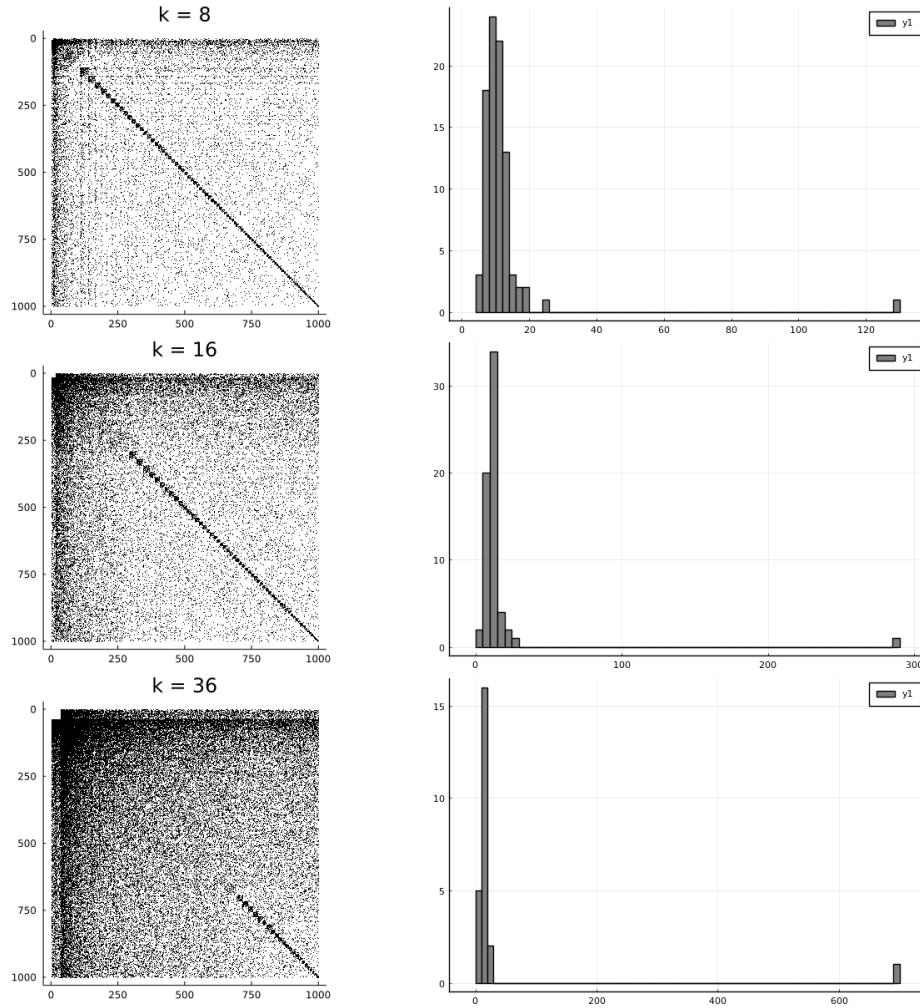
Adjacency matrix view for various values of k, b and n = 1000.

By increasing **b**, the average path length decreases rapidly, which can be seen by the "fuzziness" of the adjacency matrix. In contrast increasing **k** leads to fewer but larger communities, which can be seen by the larger dark blocks and the horizontal axis of the histograms.

7 The scale-free model

The scale-free model instead aims to create random networks with the degrees of its nodes following a power law, a property that many real world networks possess. The idea behind the power law is that, when a new node is introduced to an existing network, it is more likely to connect to an already highly connected node. A real life example would be writing a paper on networks. It is way more likely that this paper will reference Pál Erdős, arguably the most well known researcher in the field, rather than someone relatively unknown. This leads to the creation of hubs, a small number of nodes that have a disproportionately higher degree than the rest. The Barabási–Albert model using these assumption starts with a preexisting network and adds a number of nodes to it. Each new node randomly connects to other ones, however the higher the degree of a node the larger the probability a new node connects to it. To analyze this model we choose to start with **k** nodes, growing the network until it reaches a size of 1000, connecting each new node to **k** preexisting ones.

Adjacency matrix view for various values of k and n = 1000.

The preferential attachment mechanism is clear from the matrices above. Lines with higher density correspond to hubs, meaning that the particular node has a high degree. As **k** increases, more hubs are created. Upon studying the histogram we notice that most communities, independently of k, have a size of around 15 nodes. Additionally, a very small number of "hyper-communities" is formed. Thus the clustering property of the small-world model is lost.