Network Analysis
Sending information through a network

Samuele Crea

June 2024

1 The Study

In this Assignment we discuss how a piece of information can spread within a real graph in a gradual way, moving from one node to another through edges.

The idea of the experiment lies in the fact that, at the beginning, a number of **Gossipier** that contain the information are chosen and these send it to the others with a "contagion" (threshold) rate that varies from simulation to simulation.

For simplicity, Gossipiers will be represented by the color green while nodes not yet "informed" by the color blue.
There are also red-colored nodes, i.e., "malicious" nodes that contain false information that differs from the original information.

The simulation can end in different ways (especially depending on the threshold chosen):

1. information reaches all possible nodes
2. the spreading of information comes to a stop because gossipers fail to reach the threshold of "contagion"

It is interesting in our study to see how the information spreads in the "real" graph and what role malicious nodes play.

2 The Graph

For the study, I used the same graph as in the first assignment, the one representing the GPlus social network connections.
However, the latter was really too large, so that I could not test the simulation of information transmission.

I decided to reduce the graph through a **snowball sampling**.
This type of sampling starts with an initial set of nodes and recursively adds neighbors until the desired number of nodes is reached.

The initial set of nodes from which I started the sampling is not random, in fact I chose the most important nodes based on degree, betweenness and closeness centrality characteristics.

('2300', '8306', '1876','2622') is thus the set of nodes from which the algorithm started, going up to the value of 6000 nodes (a quarter of the initial set).

Through this type of sampling it was possible to greatly reduce the graph but at the same time keep all its metrics unchanged or similar.
The three figures below show the main metrics of the graph.

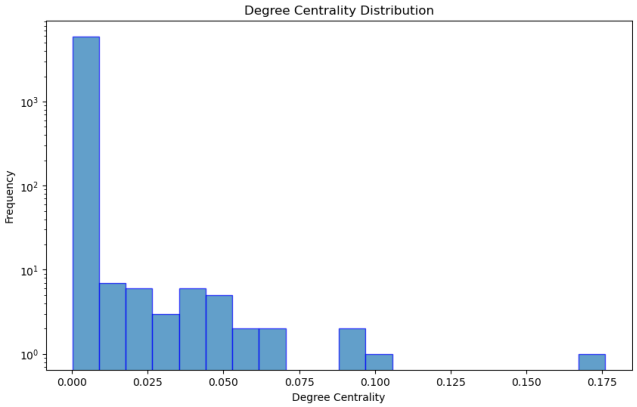

Figure 1: Degree centrality

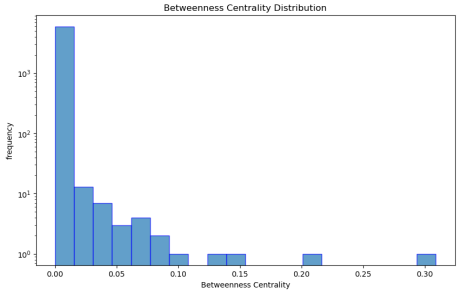

Figure 2: Betweenness centrality

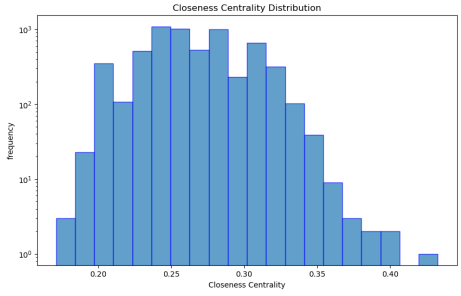

Figure 3: Closeness centrality

As can be seen from the graphs, the metrics have remained very similar to the initial graph, except of course the size of the nodes has decreased dramatically.

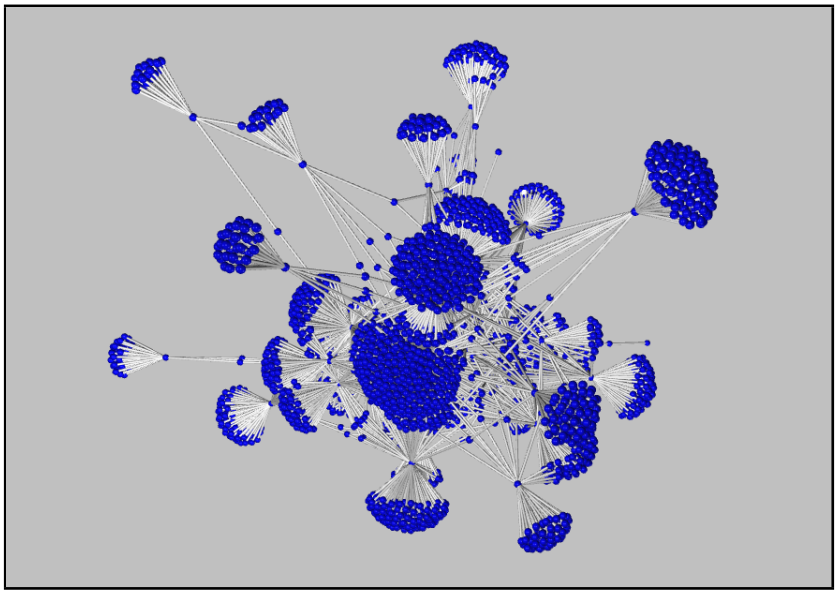

Figure 4: A better visualization of the graph

As we can see from the figure, our graph maintains a similar structure to the starting one and is only more "bare" than the original one.

With this result it is now possible to begin our investigation of information transmission in our graph.

3 Sending information through a network

3.1 Overview

As mentioned above, the simulation starts with all nodes colored blue, which means that they do not possess any kind of information.
Then nodes are chosen in the graph **Gossipier** and nodes **Malicious**, the former green and the latter at first gray and later red.
The role of Malicious nodes is peculiar: at first these are for all intents and purposes "uninformed" (gray) nodes like all others, but the moment a gossipier informs them, they become malicious (red) nodes that communicate modified information.

As will be seen then, the initial appearance of the graph is that of an almost completely blue mass that slowly, step by step, becomes green and partly red.

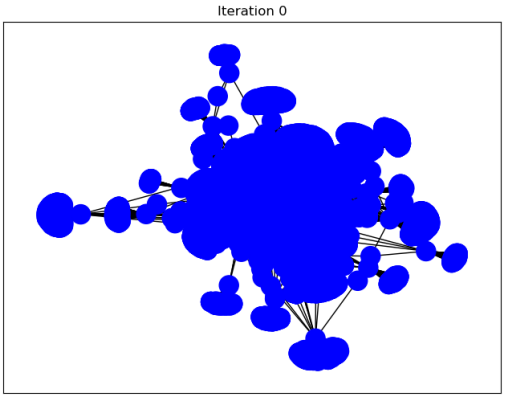

Figure 5: Initial Graph

3.2 Parameters

The parameters used by our simulation are mainly 3:

- threshold: is a value between 0 and 1, represents the threshold of information transmission. In other words, it is the proportion of informed neighbors required for an uninformed node to change state and become informed.
- num gossipier: Is the initial number of nodes that contains the information and tries to disseminate it
- num malicious: is the initial node number that contains malicious information and will try to spread it after it is informed by a gossipier.

In addition each node will be associated with an "information" which for gossipiers will be the string "Hello world!" while for malicious nodes will be a version of the same string that however differs by one letter.

The threshold value has its function primarily in one of the functions, the one used to transmit the information.
During each iteration of the information dissemination cycle, each uninformed node checks the proportion of neighbors that are Gossipier.

If the proportion of informed neighbors exceeds the threshold, then the uninformed node changes state: if the majority of informed neighbors are malicious, then the node becomes malicious otherwise the node becomes gossipier.

3.3 Test with random gossipier

Wanting to test different situations within the graph, I decided to run the script with different parameters.
A first simulation was run with an extremely small threshold (0.01), which allowed each round to send the information at a safe hit.

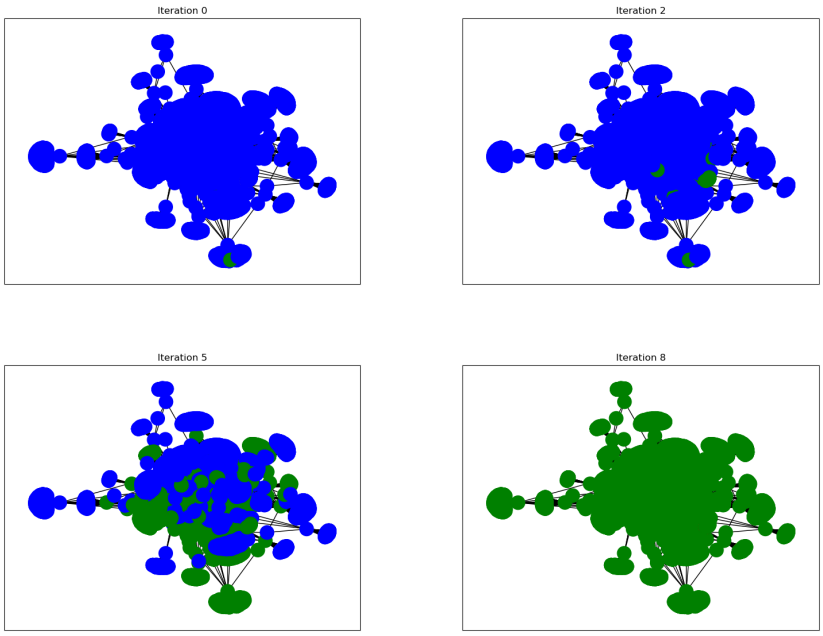

Figure 6: Spreading of the information

In this simulation, the number of Gossipier and Malicious nodes is really low: 10 and 8, respectively.
Despite this after 8 rounds, accomplice to the really low threshold, the whole graph was correctly informed.
It is interesting how the number of malicious nodes increased little during the information dissemination, probably because as we can see the gossipiers were concentrated in a small component and the malicious nodes were far away from them.

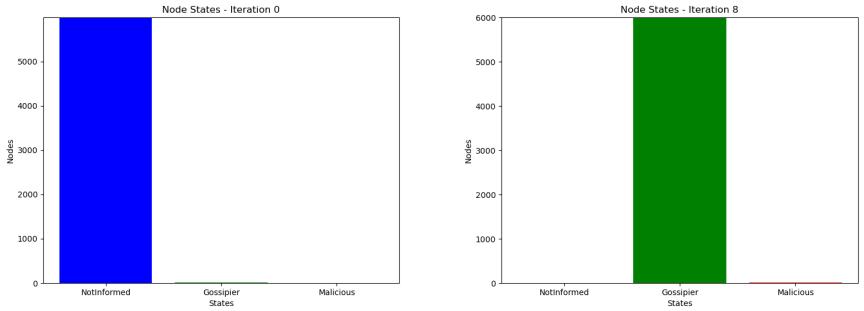

Figure 7: Number of Nodes

From the graph in Figure 7, it is better seen how the number of Malicious Nodes goes up, even if slightly, during the simulation.

Moving to a more real-world test, I decided to use a more sensible number of gossipier and malicious, 600 (10 percent of our graph) and 500, respectively.

I then decided to repeat the experiment with two different thresholds: 0.1 and 0.5

As we can see in figure 8 below, despite the rather low transmission threshold, the information could not be spread throughout the graph.

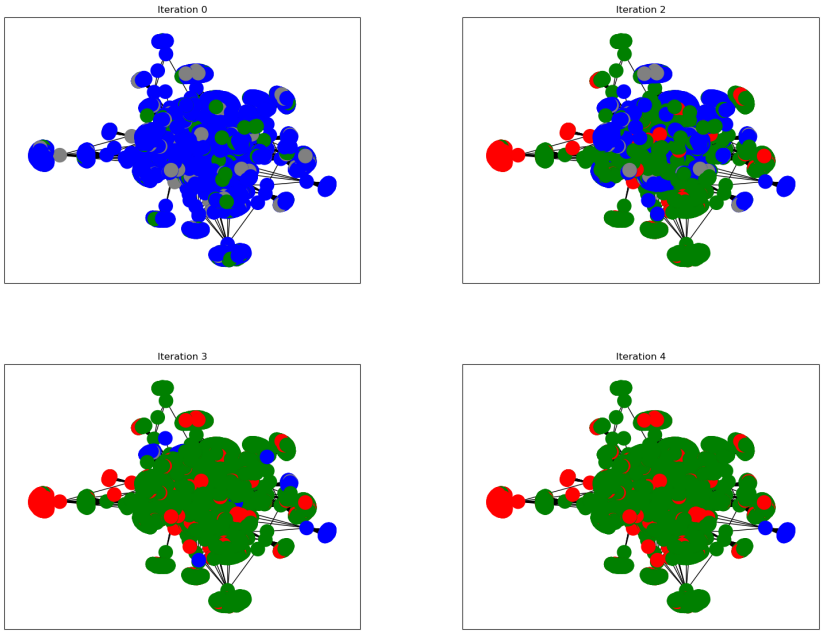

Figure 8: Spreading of the information

It is interesting how this time the spread of false information was much more pronounced, resulting in whole components of the graph such as the leftmost one.

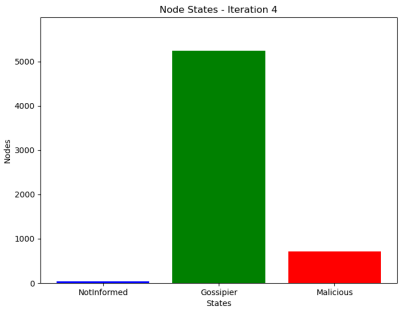

Figure 9: Number of Nodes

As we can see from the situation at the end of the diffusion , this time the malcious nodes increased significantly compared with the first experiment.

In the second experiment, a threshold of 0.5 was used, making diffusion more difficult.

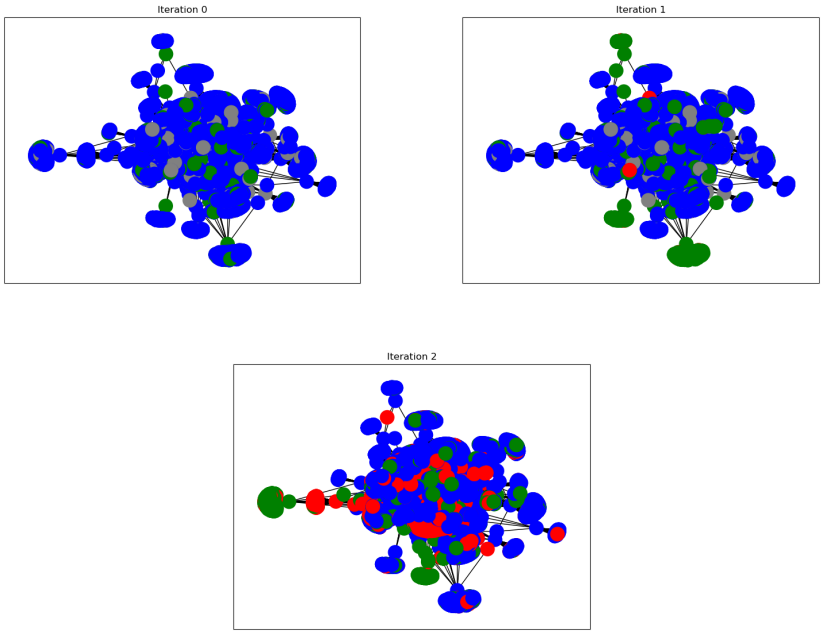

Figure 10: Spreading of the information

As was to be expected, with a higher threshold the gossipers struggled much more to spread the information and then stopped only after 3 iterations.
The result in Figure 10 shows how most of the graph remained uninformed.

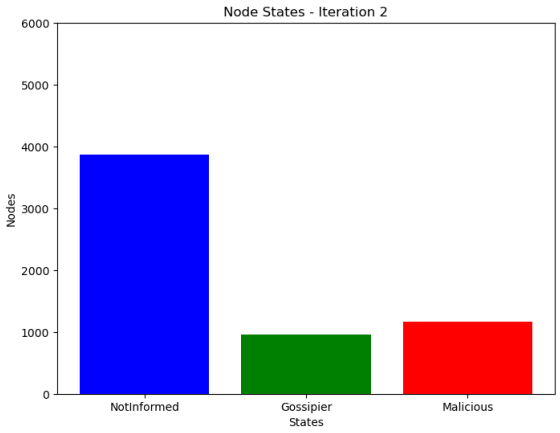

Figure 11: Number of Nodes

Another piece of information that transpires through from the graph in Figure 11 is that, in this case, the number of final malicious nodes exceeded the number of "good" gossipiers.
This may be a random fact, but it is still interesting to report.

3.4 Test using a choosen node

One more test I decided to do was with a single node and zero gossipier.

This type of test would not be significant if it were not for the choice of node, which is 2300.
This node was found to be the most influential in the graph from the analysis done in the first assignment.
Node 2300 is the one with the most links and has the highest degree, betweenness and closeness centrality.

Let us now look at how it behaves as the only "vector" of information, with a threshol of 0.1 (small but as we have seen not too much).

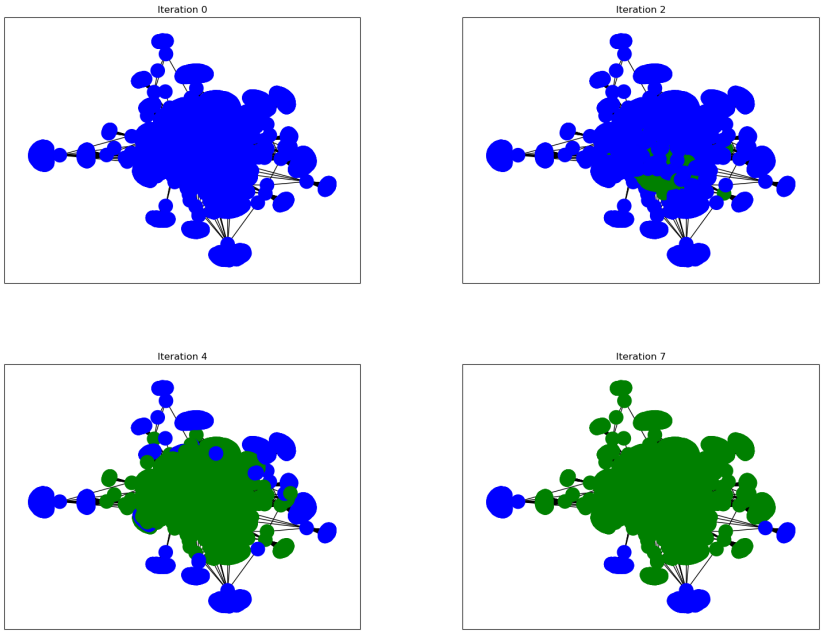

Figure 12: Spreading of the information

As we can see despite the presence of only one gossipier the dissemination of the message was almost completed, precisely because node 2300 is by far the most connected and influential node in the graph.

3.5 Cosine similarity

Cosine similarity is a measure of similarity between two vectors in a multidimensional space.
It is calculated as the cosine of the angle between the two vectors.

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

In the context of our assignment, cosine similarity is used to measure the similarity between the information (strings) known by the nodes in the network.
The textual information is transformed into TF-IDF vectors representing the frequency and importance of words.

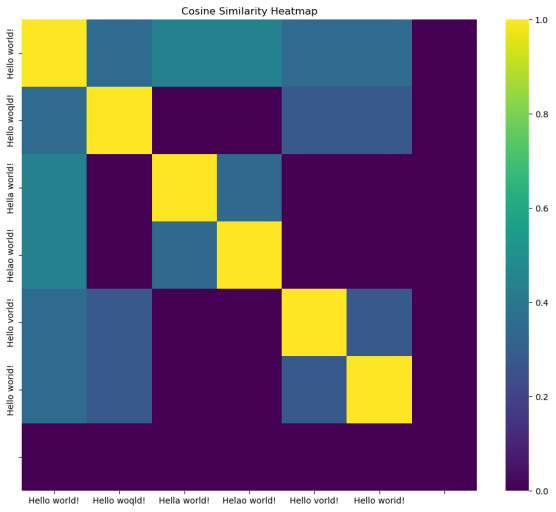

Figure 13: Cosine Similarity

The cosine similarity in the figure is that of the experiment with 600 gossipiers and threshold of 0.1.

The phrases "Hello world!" were recognized as identical, showing the highest similarity between them.
This means that every node that knows this phrase will have an exactly identical vector representation.

Sentences with typographical errors are considered less similar to the original sentence, as indicated by the lower cosine similarity values, indicating that nodes that know these variants have vector representations less similar to "Hello world!".

This type of representation allows for a clear visualization of how information has spread across the graph and how variations in information affect similarity between nodes.