Lab6: Dynamic processes over graphs

Liu Qingqing s315203
Politecnico di Torino

I. Introduction

The primary objective of this experiment is to conduct a simulation study using a voting model, exploring the probability and required time to achieve consensus under different parameters. Specifically, we will design experiments in two parts:

Part 1 - Impact of Initial State Probability: By adjusting different initial state probabilities p_1, we aim to observe the probability and time required for the system to reach consensus under various initial conditions. This helps us understand the influence of the initial state on consensus formation and the potential scenarios in different social or network contexts.

Part 2 - Influence of grids on Consensus: Keeping the initial state probability $p_1 = 0.51$ constant, we investigate the impact of different grids k on the consensus formation of the system. By altering the dimensionality of the network, we can simulate various types of network structures, providing deeper insights into the influence of network topology on consensus.

II. Methodology

A. Assumptions

We initially assume that the graph is connected, meaning that any two nodes in the graph are connected by an edge. If this assumption does not hold, we choose to study the giant connected component of the graph. This choice is made to focus on the most likely part of the actual consensus formation, while disregarding isolated components.

The initial distribution of states for each node is assumed to be independent and identical. This implies that each node initially selects a +1 or -1 state with the probability. Subsequently, at each time step, a node chooses one neighbor to update its state. As node updates follow a *Poisson process*, the time intervals between choosing neighbors and updating states are *exponentially distributed*.

B. Data structures

- **Graphs** : are represented using the NetworkX library, including nodes and edges of the graph.
- **Matrices** : Adjacency matrix represented using the lil_matrix from the SciPy library.
- **Dictionaries** : are used to store information such as the state of nodes and their neighbors.
- **Lists** : are used to store information like state probabilities, time lists, and more.
- **Tuples** : are used for representing coordinates or node-related information.

C. Voter Algorithm

Algorithm 1 *

Voter Model Simulation

Input: Graph G, Initial states of nodes, Simulation time $t = 0$
Output: Converged state of nodes
while not converged **do**
    Select a random node v
    Select a random neighbor w of v
    Upon wake-up at time t_v, where t_v is drawn from exponential distribution with rate $\lambda = 1$
        Set $X_v(t_v^+) = X_w(t_v^-)$ {i.e., v copies the state of w}
    Update the simulation time t to $t + t_v$
end while=0

D. Input

- **Probability** $p_1 = [0.51, 0.55, 0.6, 0.7]$: The probability of the initial state being +1.
- **Number of Nodes** $n = 1000$: The total number of nodes in the network.
- **Simulation Count** $sim\_time = 20$: The number of simulations to run, used to obtain statistical data on consensus probability and required time.
- **Dimension** $k\_values = [2, 3]$ (used only in Part 2 of the experiment): If the second part of the experiment is conducted (impact of dimension on consensus), this represents the dimension of the network.

E. Ouput

- **Probability**: The probability of reaching consensus as +1. For the first part of the experiment, it is the consensus probability at different p_1 values; for the second part of the experiment, it is the consensus probability at different dimensions k.
- **Time**: The average time required to reach consensusn as +1. Similarly, it is the average time at different p_1 values or different dimensions k.

III. Experiment

A. (Part 1): Impact of Initial State Probability

1) Initialize Graph: Utilized the NetworkX library for graph representation, including nodes and edges. Implemented a sparse matrix using SciPy's `lil_matrix` to succinctly store the adjacency matrix, facilitating the storage of node connections.

2) Initialize Node States: Applied the voter model by randomly generating node states (+1 or -1) based on the probability p_1 and stored the initial node states in a dictionary for reference.

3) Find Neighbors: Developed a function to find neighbors for each node, considering the graph's connectivity. If the graph was not connected, the analysis focused on the largest connected component. The information about neighbors was stored in a dictionary.

4) Simulation Loop: Employed the `initialize_graph` function to generate a connected graph and, if necessary, extracted the giant connected component for analysis. Implemented the voter model updating mechanism: at each time step, a node was randomly selected, and its state was updated to match that of a randomly chosen neighbor. This process was repeated until consensus was reached.

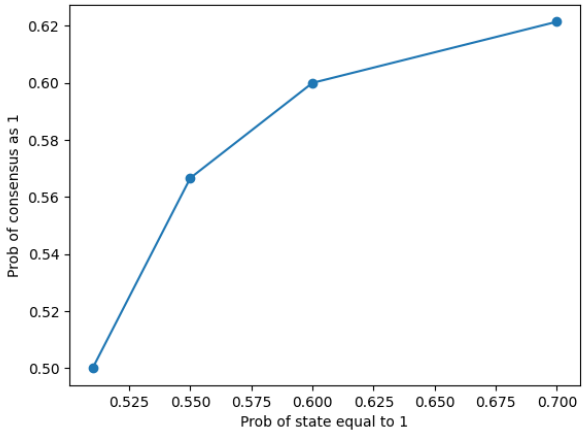

Fig. 1. Probability of Consensus as 1 vs. Probability of State Equal to 1

B. (Part 2): Different grids

1) Initialize Graph: We utilized the `k_grid` function to initialize graphs with different dimensions for the second part. Here, s represents the number of nodes in each dimension. For example, a 31x31 grid was generated, totaling 961 nodes, for the two-dimensional case. Similarly, a 10x10x10 grid was generated, totaling 1000 nodes, for the three-dimensional case.

2) Find Neighbors: For each node, a nested loop is used to iterate over possible neighbors. `itertools.product([-1, 0, 1], repeat=k)` generates all possible combinations in a k-dimensional space, representing potential movement directions for each dimension. For each potential neighbor, its coordinates are determined by calculating the sum of the current node's coordinates and the movement direction. Subsequently, it is ensured that each coordinate of the neighbor is within the range [0, size), and the neighbor is not equal to the current node.

3) Simulation Loop: Similar to Part 1.

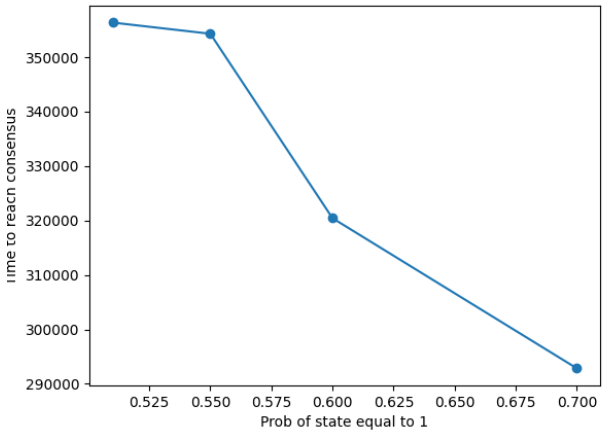

Fig. 2. Time to Reach Consensus vs. Probability of State Equal to 1

TABLE I
IMPACT OF INITIAL STATE PROBABILITY P1

p_1	Probability of Consensus as 1	Time to Reach Consensus
0.51	0.5055	359875.52
0.55	0.5142	355653.52
0.6	0.5735	319994.61
0.7	0.7250	292109.58

C. Stop condition

For each node, when it reaches consensus, we record the time and proceed to the next node to search for consensus.For every 20 nodes evaluate the probability and time. Then we set a maximum of 30 simulations for each graph. Starting from the third simulation (to have statistical significance), we calculate accuracy using a confidence interval. If the accuracy exceeds 0.9, the simulation automatically stops.

D. Results

1) Part 1: In Table I, as p_1 increases from 0.51 to 0.7, the probability of reaching consensus as +1 generally **increases** (Fig. 1) and almost equal to p1. The time to reach consensus generally **decreases** with an increase in p_1 (Fig. 2).

The simulation results suggest that the initial state probability (p_1) has a significant impact on both the probability of reaching consensus and the time required for consensus formation. And the probability of reaching consensus is almost equal to p1 for each.

2) Part 2: In Table II ,as k changes from 1 to 2, there is almost no change in the probability of reaching consensus as +1 (Fig.3) . However, the time to reach consensus significantly and even doubles (Fig.4), indicating that grids of different dimensions possess distinct graph structures, influencing consensus time. When k transitions from 2 to 3, the probability of reaching consensus as +1 remains constant, but the time to reach consensus is shorter than $k = 2$. This is attributed to $k = 3$ nodes having more neighbors, facilitating faster consensus in finite graphs.

The variation in k for grids appears unrelated to the probability of reaching consensus, which remains equal to the value of p_1. This validates the theoretical values. However, for grids with k greater than or equal to 3, the real-world computation of their consensus-reaching situation in an infinite graph is infeasible, preventing the verification of these theoretical values.

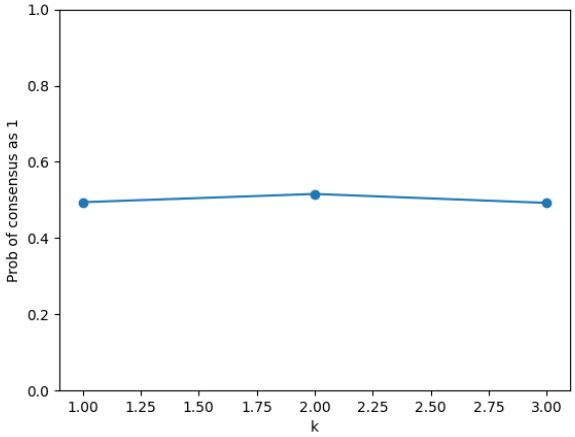

Fig. 3. Effect of k-grids on Consensus Probability

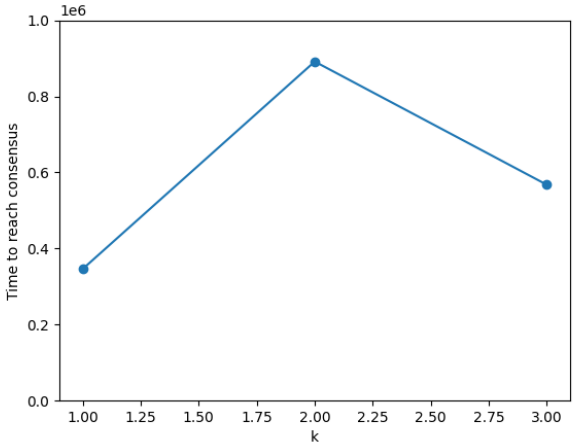

Fig. 4. Effect of k-grids on Time to Reach Consensus

TABLE II
EFFECT OF K-GRIDS

k (p1=0.51)	Probability of Consensus as 1	Time to Reach Consensus
1	0.5055	359875.52
2	0.5158	891456.14
3	0.4923	567556.43

IV. CONCLUSION

From the above, it can be concluded that, under finite conditions and with other factors held constant, the probability of reaching consensus is equal to the initial state probability. Furthermore, this probability is independent of the value of k. The time to reach consensus decreases as the initial state probability increases, and theoretically, this trend would continue to increase with the growing complexity of the grid in an infinite situation.

In the first part, graphs generated from matrices ultimately reach consensus. In the second part, grids with k greater than 3 in an infinite setting never reach consensus.