



Network Dynamics and Learning

Homework 3

Ali Ghorbanpour
Student ID: s309992

Niusha Parsa
Student ID: s312923

Sina Hamedani
Student ID: s315495

Introduction:

In this set of exercises, titled Homework 3, we tackle two primary types of problems. First, we will begin with an epidemic simulation problem, specifically simulating the real-world H1N1 pandemic that occurred in Sweden in 2009, using data available from the Swedish Civil Contingencies Agency. Following this, we will report our work on a coloring problem as an application of distributed learning. The subsequent sections will delve into these topics in detail, showcasing our work step by step:

1. Influenza H1N1 2009 Pandemic in Sweden

In the first part, we will simulate the H1N1 pandemic that occurred in Sweden in 2009. This problem is divided into four main sections. In Section 1, we will begin with some preliminary steps, such as simulating the epidemic on a k-regular graph and then generating a preferential graph. In Section 2, we will simulate the pandemic assuming no vaccination has taken place. In Section 3, we will introduce vaccination into our scenario to observe its impact on controlling the pandemic. Finally, in Section 4, we will compare our simulated data to real-world data to determine the best parameters that represent the actual situation.

1.1 Preliminary parts

1.1.1 Epidemic on a known graph:

In this section, we simulate an epidemic on a **k-regular undirected graph** representing our society. **Each node symbolizes an individual, and each edge depicts a real-world connection between two individuals.** We employ the SIR model to track epidemic data, with S, I, and R representing **Susceptible**, **Infected**, and **Recovered**, respectively.

The likelihood of infection is given by the equation $P(X_i(t+1) = I | X_i(t) = S) = 1 - (1 - \beta)^m$, where m is the count of a node's infected neighbors.

RO

In our model, the parameter ρ ranges from 0 to 1, indicating the probability that an infected node (state **I**) will recover (state **R**) during a time step, denoted by $P(X_i(t+1) = R | X_i(t) = I) = \rho$. For our simulation, we consider a network with $|V| = 500$ nodes, setting $k = 4$, $\beta = 0.3$, and $\rho = 0.7$. We simulate the spread of the **SIR** infection, initially infecting **10 randomly chosen nodes**. This process is repeated $N = 100$ times to observe the transitions of nodes among the states S, I, and R. Throughout each simulation, we record the number of susceptible nodes that become infected at each time step. After completing the simulations, we

'beta': refers to the infection rate, which represents the probability of an infected node transmitting the infection to a susceptible neighbor.

calculate the average behavior of the nodes by taking the mean results from the 100 iterations.

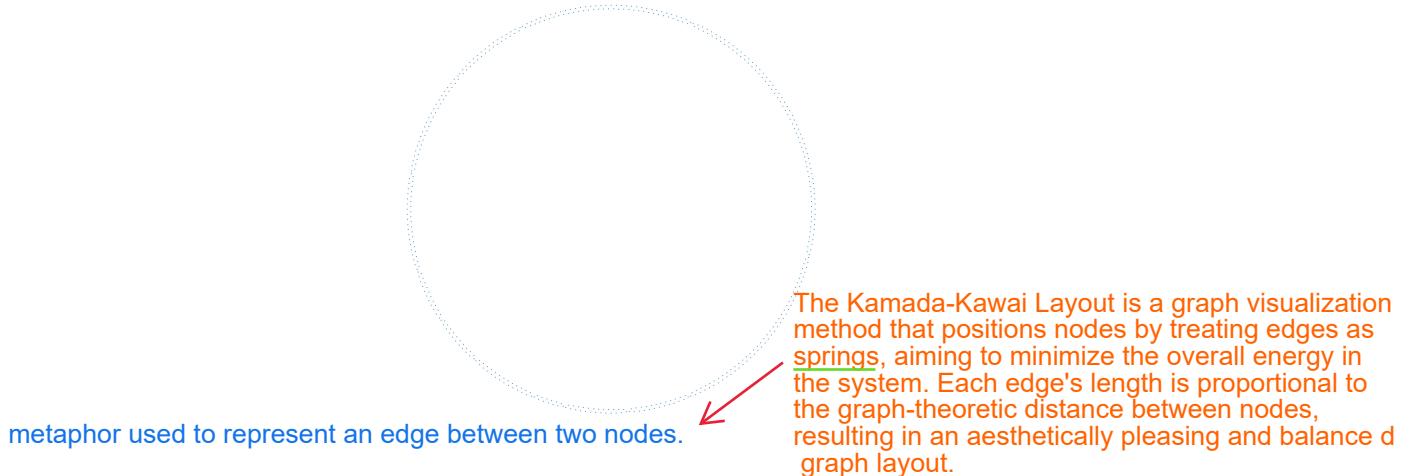
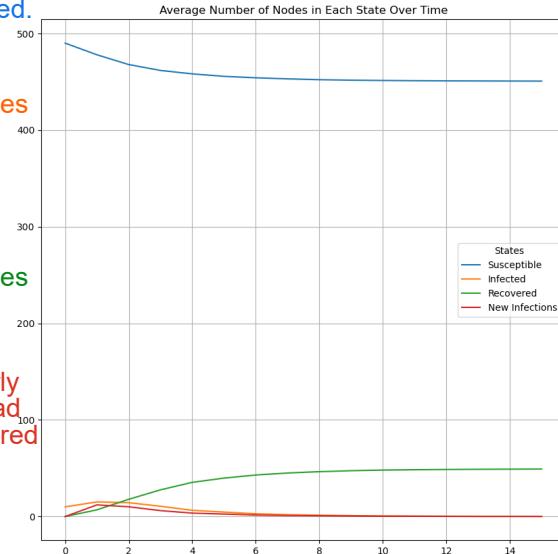


Figure 1: Visualization of a 4-Regular Graph with 500 Nodes Using Kamada-Kawai Layout.

(Blue): The number of susceptible nodes decreases over time as they become infected.



(Orange): The number of infected nodes rises initially but then declines as infected nodes recover.

(Green): The number of recovered nodes increases steadily, indicating that more nodes are recovering from the infection over time.

(Red): The rate of new infections peaks early and then decreases, showing that the spread slows down as more nodes become recovered and fewer susceptible nodes remain.

Overall, the epidemic starts with a rapid increase in infections, followed by a transition to recovery, eventually leading to a stabilization where most nodes are either recovered or still susceptible, with few active infections.

Figure 2: Epidemic Simulation: Average Node States Over Time.

1.1.2 Generate a random graph:

In this section, we employ the Preferential Attachment Model to generate a random graph. This model allows us to create a graph with a desired degree k and a specific number of nodes. The process begins with a $(k + 1)$ -complete graph, and at each step, a new node is added and connected to $k/2$ existing nodes, continuing until the graph reaches the desired number of nodes n .

The probability of forming a connection between a new node n_i and an existing node i at timestep t is determined by the degree of node i at the previous timestep $t - 1$, relative to the sum of the degrees of all

nodes at $t - 1$. Formally, this probability is given by:

$$P(W_{n_i,t} = W_{i,n,t} = 1 | G_{t-1} = (V_{t-1}, E_{t-1})) = \frac{w_i(t-1)}{\sum_{j \in V_{t-1}} w_j(t-1)}$$

Thus, the structure of the graph is influenced by the chosen values of k and the final number of nodes n .

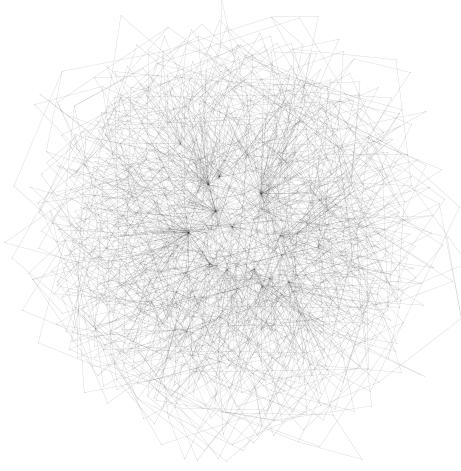


Figure 3: Visualization of a Preferential Attachment Graph with 900 Nodes Using Kamada-Kawai Layout.

1.2 Simulate a pandemic without vaccination

Here, we employ the Preferential Attachment model and the SIR model to simulate the pandemic scenario without vaccination. The initial conditions for the simulation are as follows:

$$|V| = 500 \text{ nodes}, k = 6, \beta = 0.3, \rho = 0.7, \text{infected nodes} = 10, 15 \text{ weeks}, 100 \text{ simulations}$$

Initially, the number of susceptible nodes (blue) decreases rapidly as they get infected. The number of infected nodes (orange) spikes early, indicating a rapid spread of the infection, but then declines as nodes recover. The recovered nodes (green) increase steadily and dominate the population after the initial weeks, showing that most of the population recovers over time. The new infections (red) peak early and then drop, indicating that the infection spread slows down significantly as more nodes recover and fewer susceptible nodes remain. Overall, the epidemic stabilizes with a high number of recovered nodes and very few new infections or active cases.

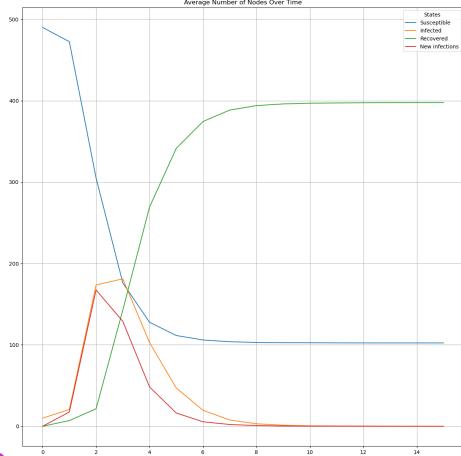


Figure 4: Average number of susceptible, infected, recovered nodes, and new infections over time.

As evident from the plot above, the pandemic will eventually come to an end even without vaccination. In the next step, however, we will examine the impact of vaccination on controlling the pandemic.

1.3 Simulate a pandemic with vaccination

Similar to the previous step, we will use the Preferential Attachment model and the SIR model to simulate the pandemic. However, in this step, we will control the pandemic by vaccinating 60 percent of the population. The initial assumptions for the simulation remain the same as in the previous step, including:

$$|V| = 500 \text{ nodes}, k = 6, \beta = 0.3, \rho = 0.7, \text{infected nodes} = 10, 15 \text{ weeks}, 100 \text{ simulations}$$

$$Vacc(t) = [0, 5, 15, 25, 35, 45, 55, 60, 60, 60, 60, 60, 60, 60, 60]$$

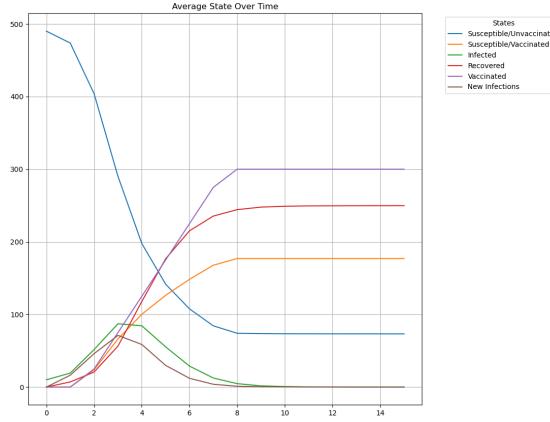


Figure 5: Average state transitions during an epidemic with vaccination over 15 weeks.

As observed, the number of infections has decreased due to vaccination. However, in a larger population and broader society, the impact of vaccination would be more pronounced.

1.4 The H1N1 pandemic in Sweden 2009

This section takes a different approach from the previous parts. Here, we will simulate the outbreak with various initial values, aiming to compare the results with real data. Our goal is to identify the best parameters k , β , and ρ that most closely match the real data, thereby demonstrating which simulation parameters are closest to reality. The values of the parameters are scaled down by a factor of 1/10,000, meaning that one simulated individual represents 10,000 real people. The evaluation of our algorithm is based on Stochastic Gradient Descent (SGD).

In this method, we start with initial values $k = k_0$, $\beta = \beta_0$, and $\rho = \rho_0$. We then test all possible combinations of these parameters, observe the simulation results, and calculate the Root Mean Square Error (RMSE) to determine which set of parameters to adjust (either adding or subtracting a Δ value). The RMSE between our simulation data and the real data is calculated using the following formula:

$$\text{RMSE} = \sqrt{\frac{1}{15} \sum_{t=1}^{15} (I(t) - I_0(t))^2}$$

Here, β_0 , ρ_0 , and k_0 are the initial values we specify, and during testing, we adjust these values by β_Δ , ρ_Δ , and k_Δ . In this context, $I(t)$ represents the number of recognized infected persons per week in the real world (scaled to 10,000). The following real-world data has been used for our simulation:

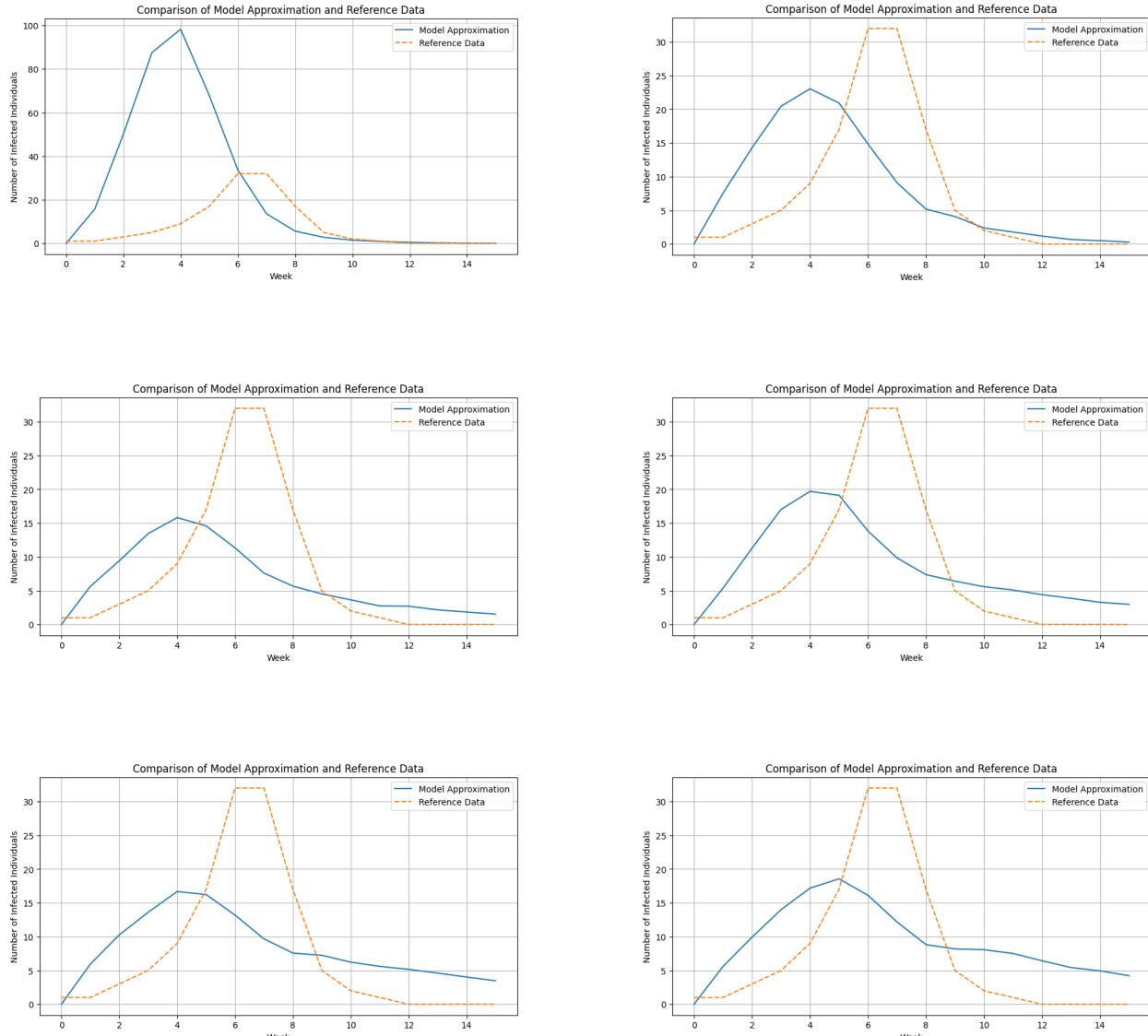
Comparison Plots: The comparison plots show the number of infected individuals over time, where the solid blue line represents the model's approximation, and the dashed orange line represents the reference data. The goal is to have the blue line closely follow the orange line.

Observation: The plots indicate that the model's approximation closely follows the trend of the reference data, especially in capturing the peak of infections and the subsequent decline.

$$|V| = 934 \text{ nodes}, k_0 = 6 \text{ initial}, \beta_0 = 0.3, \rho_0 = 0.6, \text{infectednodes} = 10, \text{weeks} = 15, \text{100simulations}$$

$$Vacc(t) = [5, 9, 16, 24, 32, 40, 47, 54, 59, 60, 60, 60, 60, 60, 60, 60]$$

$$I(t) = [1, 1, 3.5, 9, 17, 32, 32, 17, 5, 2, 1, 0, 0, 0, 0]$$



The optimization process successfully adjusted the parameters to minimize the discrepancy between the model and the reference data, providing a good fit for predicting the spread of infection.

Using the RMSE method, we achieved the most accurate model with the optimized parameters $k = 7$, $\beta = 0.0999999999999998$, and $\rho = 0.15000000000000002$. The final RMSE value is 8.421384981106137.

2. Coloring

In this section, we will explore two coloring problems as an application of distributed learning. The primary objective is to assign colors to the nodes of a network such that no two adjacent nodes share the same color.

2.1 Part 1

In the first part, we generate a 10-node graph with nodes initially colored red. At each time step, a node is selected uniformly at random, and a probability function is used to determine the color assignment as follows:

$$P(X_i(t+1) = a | X(t), I(t) = i) = \frac{e^{-\eta(t) \sum_j W_{ij} c_{j,a}(t)}}{\sum_{s \in S} e^{-\eta(t) \sum_j W_{ij} c_{j,s}(t)}},$$

Here, the cost function η sums up the number of neighbors with the same color for each node, defined as:

$$c(s, X_i(t)) = \begin{cases} 1 & \text{if } X_i(t) = s, \\ 0 & \text{otherwise.} \end{cases}$$

The noise function $\eta(t)$, which increases over time, is given by:

$$\eta(t) = \frac{t}{100}$$

To measure whether the model has achieved an acceptable result, we use a potential function that calculates conflicts. As shown in the figures, we reached equilibrium (no conflict) after approximately 480 iterations.

$$U(t) = \frac{1}{2} \sum_{i,j \in V} W_{ij} c(X_i(t), X_j(t))$$



Figure 6: Graph nodes colored in alternating orange and light blue.

The "Potential Function over Time" plot illustrates how the potential $U(t)$ of the graph evolves as the node colors are updated. The potential function measures the number of adjacent nodes with the same color, which the simulation aims to minimize over time.

Initially, the potential is high, indicating many adjacent nodes share the same color. As the simulation progresses, the potential decreases, showing that the graph is becoming more optimally colored with fewer adjacent nodes having the same color. The plot shows fluctuations in the potential due to the stochastic nature of the node color updates, but overall, it trends downward, demonstrating the system's progress towards a lower-energy state. The end of the simulation shows a steady potential, indicating that the graph has reached a more stable configuration.

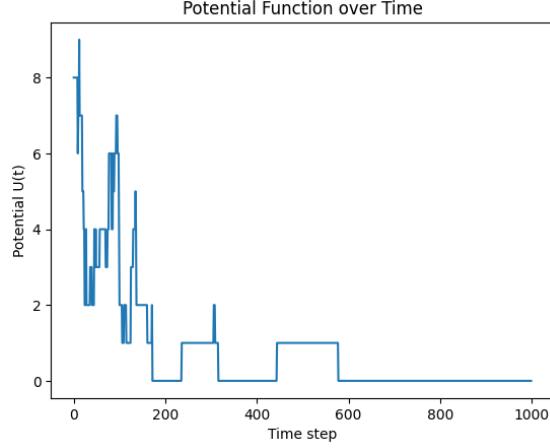


Figure 7: Potential function over time showing the convergence of the system.

2.1 Part 2

In this section, we approach a similar problem as before but with a few modifications. This time, we have 100 Wi-Fi routers with interconnections among them. The routers wake up based on a uniform random distribution, and their frequencies are assigned using the probability function from the previous problem. Instead of red and green, we now have 8 colors: $C = \{1 : \text{red}, 2 : \text{green}, 3 : \text{blue}, 4 : \text{yellow}, 5 : \text{magenta}, 6 : \text{cyan}, 7 : \text{white}, 8 : \text{black}\}$. The noise function remains the same, but the cost function has been updated as follows (same frequency costs 2, one distance costs 1):

$$c(s, X_j(t)) = \begin{cases} 2 & \text{if } X_j(t) = s, \\ 1 & \text{if } |X_j(t) - s| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The objective is to simulate this network with the specified parameters. The result of the potential function is shown below. It is evident that after approximately 600 iterations (waking up and specifying), the potential function value reaches nearly its minimum, which is 6. This indicates that after a sufficient number of iterations, the network reaches a nearly stable state with a minimal number of conflicts, amounting to 6.

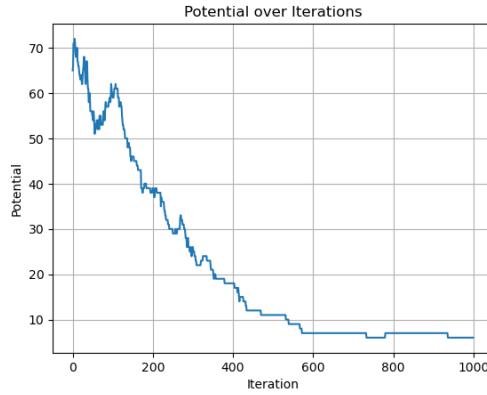


Figure 8: Potential function showing stabilization over iterations.