Luis Kligman

Applied Linear Algebra Lab

December 3rd, 2025

Markov Chains

A Markov chain is a mathematical system that lets us move between states that have certain probabilities, where the next state only depends on the current state, excluding the previous states. We can set this up by listing all of the possible conditions the system can be in, building the transition matrix and the initial distribution vector, then solving using matrix multiplication. As we try and find states over multiple iterations, we maintain the initial state but multiply the transition matrix by itself as each day or state change takes place. For the first problem, we first define the transition matrix P, inserting all of the given values that correlate the likelihood of what site someone will change to based on their current site. We make sure that each column sums to one, as this is a required property of Markov chains, probability must sum to 100 percent (being 1). For part a, we make the start matrix x0_single to denote a single employee at site 2 initially. Since the problem states 3 state changes, we can calculate x3_single by doing P^3 multiplied by x0_single. We get P to the power of 3 because there are 3 possibilities for the state to change. We can then display the 5th element of x3_single, as this will correlate to site 5 and its newfound probability. For part b of this problem, we are told we have 100 individuals uniformly distributed, which means 20 across all five sites to start. We can then reuse our previous formula, but with this new x0_pop distribution and P^4 for 4 steps. Once our result is returned, we must make sure we retain a population of 100, as in a Markov Chain, there should be no loss of population. For part c of the problem, we must find the steady state vector of our P matrix. The code first solves for the eigenvalues and eigenvectors of P using MATLAB's built-in eig()

function. We then find the index of the eigenvalue closest to 1, extract it, and normalize it to 1. As in a Markov Chain, the probabilities must sum to 100 percent (1 = 100 percent). The solution implemented in the code properly demonstrates and solves all the questions asked in problem 1. Problem 2 asks us to implement a Markov Chain to simulate a college campus of 5,000 students and how the influenza virus will spread based on certain probabilities depending on whether someone is susceptible or infected. If someone is susceptible, they have a 16 percent chance of getting the flu the next day; by extension, they have an 84 percent chance of staying susceptible. If infected, they have a 40 percent chance of recovering the next day, and a 60 percent chance of staying infected. Part a asks us to make the transition matrix between states. State 1 is susceptible, being column 1, and has values [.84; .16] denoting the probabilities of changing to the other states. Column 2 has values [.40; .60]. Once this matrix is made, we must compute each column sum and make sure they all add up to 1 to ensure the requirements of a Markov Chain. Part b asks how many students will have the flu on the second day and tenth day if we initially have only 100 students infected to start. To solve this, we know I0 = 100, and S0 = 5,000 - I0. We can then use the calculation dayN = transition_matrix^N * x0. Therefore day2 = A^2 * x0 and day10 = A^10 * x0. With these results, we must ensure the Markov Chain properties and make sure these results add up to 5,000. Once we know this, we can extract the second value in each, which correlates to the infected count. For part c, it asks how many students have the flu initially if there are 1400 students with the flu on day 3. To solve this, we can form a 2x2 linear system of equations to allow us to solve for our unknowns. This will allow us to solve for the S0 and I0 values, since we want to know our starting conditions. We use the matrix operand \ to solve the linear system Mx = b for our unknown x, which, as mentioned before, is [S0; I0]. Now we check if the S0 and I0 values sum to 5,000; if they do, we can display these results to be our proper starting conditions for this scenario.

Works Cited

Jeffrey A. Fessler and Raj Rao Nadakuditi, *Linear Algebra for Data Science, Machine Learning, and Signal Processing*, Cambridge University Press, 2024.

Charles M. Grinstead and J. Laurie Snell, *Introduction to Probability*, 2nd ed., American Mathematical Society, 1997. (See Chapter 11: “Markov Chains”.)