The Common Cold: SIS "Susceptibles vs. Infected" Model

Name: Christ-Brian Amedjonekou

Date: 2/25/2019

MAT 4880-D692 (Math Modeling II) SIS Model Project 1

Spring 2019, Section: D692, Code: 36561

Instructor: Johann Thiel

Abstract

In this study, we set out to create an SIS "Susceptibles vs. Infected" Model of the common cold over a 6-day period. The goal was to find the steady state of the dynamical system given the rate of infection, fixed population, and rate of recovery for the common cold. The common cold is a type of disease in which once recovered, a person is susceptible to infection again. Although the problem could be modeled using a discrete dynamical system, the objective required that we use a continuous dynamical system. We used sage/python to model and analyze this problem. Once modeled, finding the equilibrium points to the system is the next step as it shows the value in which the model reaches its steady state. Reaching the steady state (finding the equilibrium points) indicates that the recently observed behavior of the system will continue into the future. What that means for this problem is that we'll expect the number of cases to converge to a point and remain there, or at least this is what we believe. Finally, we were interested the reproductive number as it will tell us if the disease will spread and approach the steady state or if it will eventually reach the disease-free state. As this is an SIS model we expect to the reproductive number to be greater than 1, indicating that it will reach the steady state.

Introduction

The common cold, as the name suggests, is a very common disease that afflicts millions in the United States every year. Scientist model disease in order to determine how the disease will affect large populations, and potentially offer treatment. Infection Rates, Recovery Rates, Reproductive numbers are all important in determining how the common cold will affect large populations and determine whether treatment is necessary amongst these populations. For this report, our goal is to determine if the disease will reach the steady state; We'll determine if the disease will persist. In order to do that we create a continuous dynamical system, where we have growth rate and competition term. We'll be looking to find the equilibrium points of the system as that will determine the steady state of the system; This will be shown graphically. From there we'll look for the reproductive number as that will also determine if the disease will spread and approach the steady state.

Assumptions and Definitions

For the model of the continuous dynamical model, we'll assume the following:

$$\frac{\partial S}{\partial t} = -\frac{\beta SI}{N} + \gamma I = \beta I \left(1 - \frac{I}{N}\right)$$

$$\frac{\partial I}{\partial t} = \frac{\beta SI}{N} - \gamma I = \beta I \left(1 - \frac{I}{N}\right)$$

- where $S = Susceptibes$; $I = Infected$; $N = Total\ (Fixed)\ Population$
- where $g_S = growth\ rate\ of\ susceptibles = -\frac{\beta SI}{N}$
- where $g_I = growth\ rate\ of\ the\ infected = \frac{\beta SI}{N}$
- $c_S = competition\ term\ for\ susceptibles = \gamma I$
- $c_I = competition\ term\ for\ Infected = \gamma I$
- $\beta = Infection\ Rate$; $\gamma = Recovery\ Rate = \frac{1}{Duration}$

The Total Population, N, is a fixed population of 10000 people. The number of Infected, I, is given to be 120 members of the total population. Number of Susceptibles, $S = N - I$, is equal to 9880 members of the total population. This is reasonable due to the model not having in between cases; You're either Susceptible or Infected. Infection Rate is, β, is given to be 85% or 0.85. The Recovery Rate is, $\gamma =$

$\frac{1}{Duration}$, is assumed to be $\frac{1}{6}$ as the duration is given to be 6 days. We also assume that $S, I \geq 0$ as we cannot have negative population. We also need to know that a differential equation $\left(\text{e.g.}: \frac{\partial S}{\partial t}, \frac{\partial I}{\partial t}\right)$ simply relates the rate of change (derivate) to the state variables. To solve the model, we'll be using a computer algebra system such as Sage-math or Python with its numerous symbolic, numerical, and scientific packages.

Analysis

To solve the system, we set both differential equations to 0; Solving for the equations set to 0 gives the equilibrium points. This is because if the rate of change is 0, it means there is no change and the dynamical system has reached the steady state. Python, specifically the Sympy (Symbolic Python) package, was used to find the result for the system. $I = 0, \; I \approx 8093.22$ are the equilibrium points to the system. Other Python packages, specifically NumPy (Numerical Python) and Matplotlib (Plotting Package), were used to plot the vector plot of the system to determine which equilibrium the vectors converge to. This is shown below in Figure 1 below:
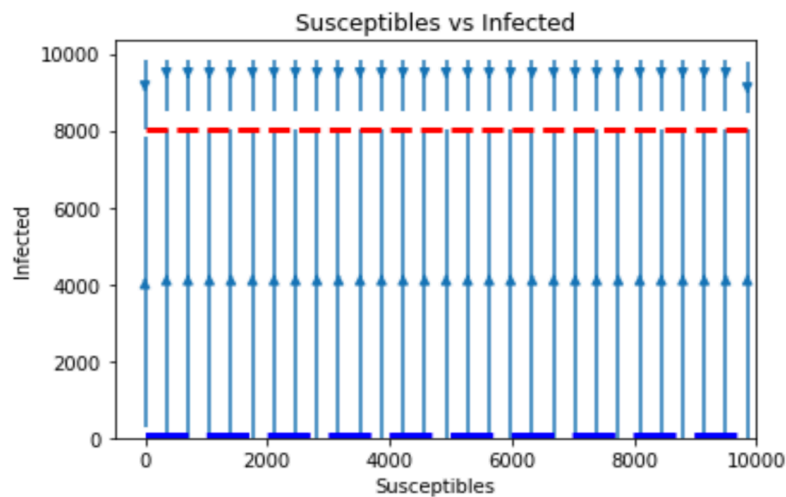

Figure 1: Vector Plot for Common Cold Dynamical System

The vector plot shows that the vectors of the dynamical system converge to $I \approx 8093.22$, which seems to indicate the infection will persist. This is further supported by the reproductive number. The value of the reproductive number determines if the disease will fizzle out or go on to reach steady state. If the

reproductive number, calculated by $\frac{\beta}{\gamma}$, is less than 1 $\left(\frac{\beta}{\gamma} < 1\right)$ then the infection will fizzle out and eventually converge to $I = 0$. However, as with this case, the reproductive number is greater than one $\left(\frac{\beta}{\gamma} \approx 5.1\right)$ so the infection will eventually converge to $I \approx 8093.22$.

Interpretations and Conclusions

Based on the results obtained from Figure 1, we can assume that the disease will reach the steady state of $I \approx 8093.22$ and stay there for future events. This is also supported by the reproductive number which is greater than 1 with a value of $\frac{\beta}{\gamma} \approx 5.1$. This shows that modeling the common cold using an SIS model was very appropriate as it is expected that no one will be truly immune; Everyone, at some point, will always be susceptible to this disease.

Appendix to the SIS Susceptibles vs. Infected Model

February 27, 2019

0.1 SIS Problem

0.2 Variables

- g_S = Growth term of the Susceptibles
- g_I = Growth term of the Infected
- c_S = Competition term of the Susceptibles
- c_I = Competition term of the Infected
- N = the total, fixed size of the population
- I = number of infected, initially
- S = number of susceptibles
- γ = Recovery Rate = $\dfrac{1}{\text{Infection Duration}}$
- β = Infection Rate

0.3 Assumptions

- $N = 10000; I = 120; S = 9880$
- $\gamma = \dfrac{1}{6}; \beta = 0.85$
- $g_S = -\dfrac{\beta S I}{N}$
- $g_I = \dfrac{\beta S I}{N}$
- $c_S, c_I = \gamma I$
- $\frac{dS}{dt} = -\dfrac{\beta S I}{N} + \gamma I = -\beta I \left(1 - \frac{I}{N}\right) + \gamma I$
- $\frac{dI}{dt} = \dfrac{\beta S I}{N} - \gamma I = \beta I \left(1 - \frac{I}{N}\right) - \gamma I$
- $S, I \geq 0$

0.3.1 Packages

```python
In [2]: import numpy as np
        import sympy as sp
        from sympy.solvers import solve
        from matplotlib import pyplot as plt
        import math as m
```

0.3.2 Variables

```python
In [3]: I = sp.symbols('I')
        gS, gI, cS, cI = -0.85*I*(1 - I/10000), 0.85*I*(1 - I/10000), (1/6)*I, (1/6)*I
        dSdt, dIdt = gS + cS, gI - cI
        sp.init_printing()
```

0.3.3 Functions

```python
In [4]: def susceptibles(infected):
            """Rate of Change (differential) for susceptibles"""
            return (-0.85*infected*(1 - infected/10000)) + (1/6)*infected


        def infected(infected):
            """Rate of Change (differential) for infected"""
            return 0.85*infected*(1 - infected/10000) - (1/6)*infected


        def reproductiveNumber(beta, gamma):
            """basic reproductive number, when (\beta)/(\gamma) > 1, the disease
            will spread and approach the second steady state; otherwise, it
            will eventually reach the disease-free state."""
            return beta/gamma

        def criticalThreshold(r0):
            """ critical vaccination threshold, which is equal to 1-(1/ R0)"""
            return (1-(1/r0))
```

0.3.4 Equilibrium Points

```python
In [5]: solve([dSdt, dIdt], I)
```

```
  Out[5]:
```

$$[(0.0),\quad (8039.21568627451)]$$

0.3.5 Plot

```python
In [10]: x, y = np.linspace(0, 10000), np.linspace(0, 10000)
         X, Y = np.meshgrid(x, y)
```

```
U, V = 0 * susceptibles(X), infected(Y)
fig, ax1 = plt.subplots()
ax1.streamplot(X, Y, U, V)
ax1.plot([0, 10000], [solve([dSdt, dIdt], I)[1][0], solve([dSdt, dIdt], I)[1][0]],
        c= 'r', ls= '--', lw= 3)

ax1.plot([0, 10000], [solve([dSdt, dIdt], I)[0][0], solve([dSdt, dIdt], I)[0][0]],
        c= 'b', ls= '--', lw= 3)
ax1.set(xlabel= 'Susceptibles', ylabel= 'Infected',
        title= 'Susceptibles vs Infected');
```
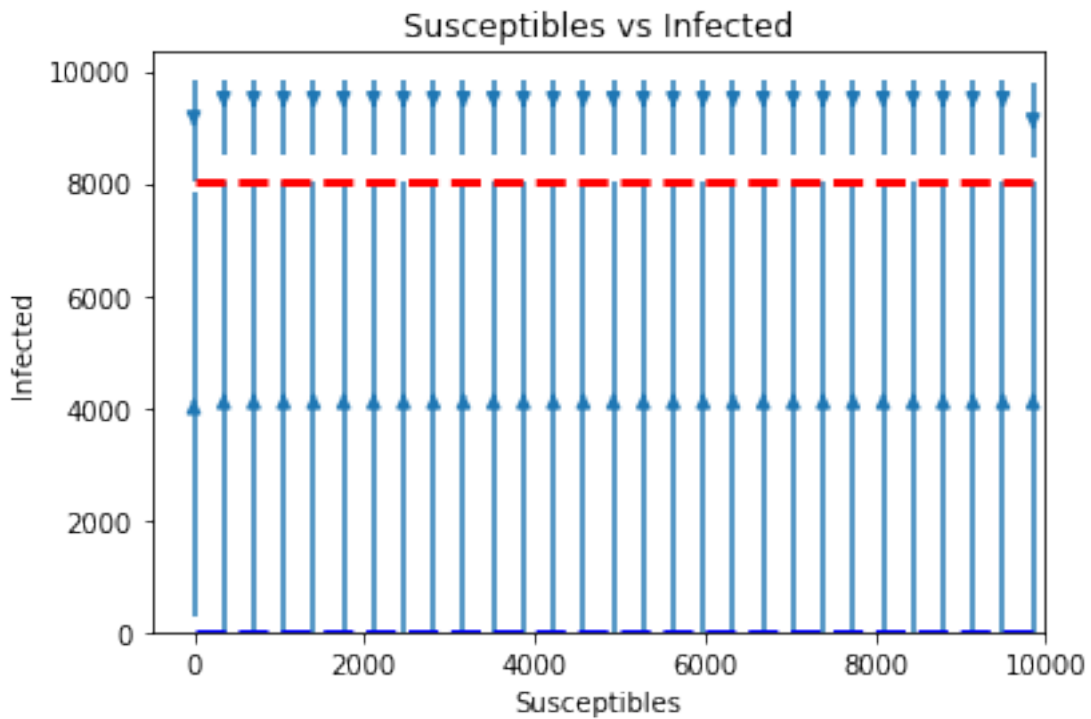

0.3.6 Reproductive Number

To calculate this value, we'll use the following formula:

- $R_0 = \dfrac{\beta}{\gamma}$

- $\gamma = \text{Recovery Rate} = \dfrac{1}{\text{Infection Duration}}$
- $\beta = \text{Infection Rate}$

```
In [6]: beta, gamma = 0.85, 1/6
        r0 = reproductiveNumber(beta, gamma)
        r0
```

`Out[6]:`

$$5.1000000000000005$$

0.3.7 Critical Threshold

`In [9]: criticalThreshold(r0)`

`Out[9]:`

$$0.803921568627451$$