Cooper, Shawn

Morris, Aaron

Matrix Theory

Dr. Cetin Savkli

3 August 2020

Final Project Paper: The Calculation of R0 on SIR Models using Matrices

**Introduction**

One of the fundamental challenges in epidemiological studies is that of determining the long-term behavior of a disease. That is, whether the disease will die out naturally or persist at endemic levels, meaning the entire population is at risk of contracting the disease. A key tool for investigating these questions as they pertain to flu-like diseases, which result in immunity or death for infected populations, is the Susceptible, Infected, Removed (SIR) model.

SIR models use a system of differential equations to track the changes in a population of size N over time as they move from the non-immune compartment Susceptible (S), to Infected (I), and either die or develop immunity and are Removed (R). Under certain assumptions, this system of differential equations can be represented as a vector-valued function. Taking the partial derivatives of the vector-valued function establishes a Jacobian transition matrix between population states. By linearizing the Jacobian and assessing the eigenvalues, we are able to determine the long-term behavior of the system as a value termed *R0*, the basic reproduction number (Diekmann, Heesterbeek, & Roberts, 2009).

Heesterbeek and Dietz describe *0* as one of the foremost and most valuable ideas that mathematical thinking has brought to epidemic theory (Heesterbeek & Dietz, 1996). *R0* is biologically understood to represent the number of secondary infections that an index case infective will cause over the course of their infectious period in a population with no previous immunity. It can be expected that *R0* > 1 will result in complete infection of a population, where *R0* < 1 will result in the disease naturally clearing from the population (Heffernan, Smith, & Wahl , 2005).

While there are multiple methods for calculating *R0 ,* this paper will demonstrate that *R0* is an alternate representation of asymptotic stability in a linearized system of equations, using real-world COVID-19 data in a simple SIR model. We then expand to discuss the methodology in more complex, multi-compartmented models*.*

**Assumptions**

The simplest SIR Models rely on several assumptions (Jones, 2007):

1. An unchanging population size N with no births or underlying death rate.
2. Constant effective contact and removal rates (β and γ, respectively).
3. A well-mixed population, implying that an infected individual has an equal chance of contacting and infecting any susceptible individual.

More complex models account for these assumptions through additional population compartments and will be addressed later.

**The Model**

In order to determine *R0*, we must first determine how the populations of each of our three compartments -- S, I, and R -- change over time. The following three ordinary differential equations form the simple SIR Model and answer this question in continuous time:

The constant β is the *effective contact rate*, which represents the probability an infected individual has of successfully transmitting the disease during each contact with a susceptible individual (Jones, 2007). The constant γ is the *removal rate* and represents the probability that an infected individual is removed from compartment I, either through recovery with developed immunity or through death.

These ODEs can be represented as a vector-valued function and taking the Jacobian allows us to visualize how each variable specifically affects each other variable in the model.

A close up of a clock

Description automatically generated

Analysis of the Jacobian is made easier at the equilibrium points of the ODEs. There are two equilibrium points in the ODEs wherein . The first is when R = N, implying that the infection has become endemic and infected the entire population. The second is when S = N, implying that the entire population is susceptible, and the infection has not yet begun. Because R0 is defined for an entirely susceptible population, we focus on the second equilibrium point, termed the *disease-free equilibrium*. Doing so linearizes the equations in terms of I.

Additionally, we are able to further simplify analysis by removing the last column and last row from our Jacobian model, as removed individuals do not affect the progression of the model. Doing so  results in the following linearized Jacobian state-change transition matrix:

The eigenvalue with the largest magnitude (Blackwood & Childs, 2018) of this matrix represents the greatest degree of variance in between compartments. In this case, the largest eigenvalue is - . Because linear systems of equations have solutions of the form [cite from class text], the system is stable when the largest eigenvalue λ is < 0.

Solving for 🡪 🡪

The biological interpretation of this result is that when /< 1, the infection will not reach endemic levels. This threshold is R0.

**Data**

A close up of a map

Description automatically generatedEpidemiological data for New Zealand was drawn from the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (Dong, Hongru, & Lauren Gardner, 2020). We selected New Zealand as the source of our data due to detailed recordkeeping, a relatively small population size, and the fact that New Zealand has seen a nearly complete epidemiological life cycle for COVID-19, by which we mean New Zealand has effectively ended the community spread of COVID-19 (Cousins, 2020).All data was calculated from the date of the first recorded infection in New Zealand to July 29th, 2020.

The data was imported to MATLAB as three vectors of daily counts for the cumulative number of confirmed COVID-19 cases, recoveries, and deaths. The New Zealand population was retrieved from the World Bank table of World Development Indicators ((JHU), n.d.; (JHU), n.d.). We established the daily number of active cases as the number of confirmed cases - recoveries - deaths, and the number of susceptible cases as the New Zealand population - the number of confirmed cases. All values were expressed as a ratio of the total population.

A close up of a device

Description automatically generatedWe then performed least-squares analysis between our observed daily counts and the SIR model ODEs in order to estimate values for parameters β and 𝞬.

**Results**

Comparison of our SIR model and the observed data from New Zealand resulted in a β value of 0.0964 and a value of 0.2142, yielding a largest eigenvalue of -0.1178, which implies asymptotic stability at the disease-free equilibrium. The corresponding R0 value is 0.4500, implying that COVID-19 will not reach endemic levels in New Zealand, as is corroborated by real-world reports.

**Extension to Multiple Compartments**

One of the underlying assumptions of the simple SIR model is that of a “well-mixed” population, which implies that each infected individual has an equal chance of infecting any susceptible individual. This is clearly not the case in a population as large as New Zealand. The effects of this assumption can be reduced by adding additional compartments to the SIR model. For instance, some diseases have asymptomatic carriers. These individuals may have a higher chance of spreading infection because they do not self-identify as sick and change their behavior. The Susceptible, Exposed, Infected, Removed (SEIR) model adds an additional compartment, which results in a 3x3 transition matrix (Ridenhour, Kowalik, & Shay, 2014). With additional data, more compartments can be added to create a more accurate model, such as age brackets [or location] where S*1* is children under 14 years, S*2* is 15 to 60, and S*3* is adults over 60, and so on. This method quickly increases the size of the transition matrices which in turn increases the complexity of eigenvalue calculation.

A close up of a piece of paper

Description automatically generated

In order to account for the increased complexity, a next generation matrix must be taken. This is first established by two matrices, T and --sometime labeled F and V. Each compartment contains these two functions; a into the compartment and a transition out of the compartment; otherwise known as F and V. In the SIR model, the single infected compartment contains the F function of SI/N and a V function of I. With many more infected compartments, these functions are not as simple.

Once the matrices that represent the F and V transformations have been identified, the next generation matrix is calculated by utilizing the formula (Diekmann, Heesterbeek, & Roberts, 2009):

NGM = T-1

After generating the next generation matrix, the largest eigenvalue of this matrix is the value of R0. Depending on the size of this matrix (i.e. the number of different compartments), the calculation of the eigenvalue can become computationally difficult at scale. This difficulty can be overcome by utilizing various methods for finding the largest eigenvalue. These methods can range from QR decomposition to the infamous PageRank algorithm. The slowest of which is the Laplace expansion calculates the exact answer in O(n!). Another popular algorithm for calculating eigenvalues, QR Decomposition can be used and calculates exact values in O(n2) computational time. Utilizing the PageRank algorithm to calculate an approximation of the eigenvalues for large matrices, the computational complexity is  O(n+m).  That said, the largest eigenvalue represents R*0* of the population with the greatest mixing, which in turn helps satisfy the basic assumptions of the SIR model.

**Conclusion**

Representing population and infection data as a vector-valued function and determining the eigenvalues of the linearized Jacobian resulted in mathematical verification of COVID-19 observational data in New Zealand. The R0 < 1 result obtained coincided with the observation that COVID-19 did not reach endemic levels. Additionally, running our model over a period of 30 days after initial infection returned similar results, implying that the model holds predictive as well as descriptive capabilities.

Increasing the number of compartments in the model increases accuracy but exacerbates computational complexity in the calculation in the calculation of eigenvalues. With an appropriately designed next-generation matrix, this increase in the complexity can be overcome by various algorithms that derive approximations of the largest eigenvalue.

Ultimately, the power of applying spectral radius analysis to complex and detailed epidemiological models is that we are able to characterize the long-term behavior of a disease when finding closed-form solutions from the underlying systems of ordinary differential equations becomes computationally inefficient.

(Cousins, 2020)

# Bibliography

Blackwood, J. C., & Childs, L. M. (2018). An introduction to compartmental modeling for the budding infectious disease modeler. *Letters in Biomathematics*, 195-221.

Cousins, S. (2020). New Zealand eliminates COVID-19. *The Lancet*, 1474.

Diekmann, O., Heesterbeek, J., & Roberts, M. (2009). The construction of next-generation matrices for compartmental epidemic models. *Journal of the Royal Society*.

Dong, E., Hongru, D., & Lauren Gardner. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet*, 533-534.

Heesterbeek, J., & Dietz, K. (1996). The concept of Ro in epidemic theory. *Statistica Neerlandica*, 89-110.

Heffernan, J., Smith, R., & Wahl , L. (2005). Perspectives on the basic reproductive ratio. *Journal of the Royal Society*, 281-293.

Ridenhour, B., Kowalik, J., & Shay, D. (2014). Unraveling R0: considerations for public health applications. *American journal of public health*, 32-41.

Appendix A - Extracting Data

function dataMatrix = loadData(table)

    %% This function loads the corona data from a git repository.

    %

    % Input:

    %    table: confirmed, deaths, or recovered

    % Output:

    %    dataMatrix: Matlab Table reflecting JHU Time Series File.

    websave([table,'.csv'], ['https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_',table,'\_global.csv']);

    opts = detectImportOptions([table,'.csv']); % Detect import parameters

    % Fix range and delimiter

    opts.DataLines = [2, Inf];

    opts.Delimiter = ",";

    opts.VariableNames(1:4) = [{'ProvinceState'}, {'CountryRegion'}, {'Lat'}, {'Long'}];

    opts.VariableTypes(1:4) = [{'string'}, {'categorical'}, {'double'}, {'double'}];

    % Fix file level properties

    opts.ExtraColumnsRule = "ignore";

    opts.EmptyLineRule = "read";

    % Fix variable properties

    opts = setvaropts(opts, "ProvinceState", "WhitespaceRule", "preserve");

    opts = setvaropts(opts, ["ProvinceState", "CountryRegion"], "EmptyFieldRule", "auto");

    dataMatrix = readtable([table,'.csv'], opts);

end

Appendix B - Estimating Beta and Gamma

clear all

confirmed = loadData('confirmed');

deaths = loadData('deaths');

recovered = loadData('recovered');

region = 171; % Queensland

t = 60:width(confirmed);

N = 4880000;

S = N - confirmed{region,t} - deaths{region,t} - recovered{168,t};

I = confirmed{region,t} - deaths{region,t} - recovered{168,t};

R = deaths{region,t} + recovered{168,t};

s = S/N;

i = I/N;

r = R/N;

exp\_y = [s(:); i(:); r(:)];

exp\_t = 1:length(i);

p0 = [0.2 0.2 60]; %initial guess for beta, gamma and the delay

p\_estimate = fminsearch(@(p)odefit(exp\_t,exp\_y,p),p0);

[~,Y] = ode45(@(t,y)odefun(t,y,p\_estimate),exp\_t,[1 1/5072000 0]);

S = Y(:,1); I = Y(:,2); R = Y(:,3);

plot(exp\_t,i,'o',exp\_t,r,'\*',exp\_t,I,exp\_t,R)

legend('i','r','I','R')

figure

plot(exp\_t,s,'o',exp\_t,S,exp\_t,)

legend('S')

function err = odefit(exp\_t,exp\_y,p)

    [~,y] = ode45(@(t,y)odefun(t,y,p),exp\_t,[1 1/4880000 0]); %initial conditions w/ 1 infected (i)

    err = sum((y(:) - exp\_y).^2);

end

function dydt = odefun(t,y,p)

        s = y(1);

        i = y(2);

       % r = y(3);

       beta = p(1); gamma = p(2); deltat = p(3);

       if t<deltat

          gamma = 0;

       end

    dydt = [-beta \* s \* i;

             beta \* s \* i - gamma \* i;

             gamma \* i];

Calculating R0

confirmed = loadData('confirmed');

deaths = loadData('deaths');

recovered = loadData('recovered');

region = 12; % Queensland;

t = 12:width(confirmed) ;

w=width(confirmed)-12 ;

qlpop = 4880000;

s1 = qlpop - confirmed{region,t} - deaths{region,t} - recovered{region,t};

i1 = confirmed{region,t} - deaths{region,t} - recovered{region,t};

r1 = deaths{region,t} + recovered{region,t};

M = transpose([s1; i1; r1]);

[b, v] = betaVegaCalc(s1, (confirmed{region,t}), (deaths{region,t}), (recovered{region,t}), w, qlpop);

s1 = qlpop - confirmed{region,t} - deaths{region,t} - recovered{region,t};

i1 = confirmed{region,t} - deaths{region,t} - recovered{region,t};

r1 = deaths{region,t} + recovered{region,t};

% Pretending Birth Rate and Death Rate do not apply.

J = [-b\*i1/qlpop -b\*s1/qlpop 0; b\*i1/qlpop b\*(s1/qlpop)-v(t) 0; 0 v(t) 0]