# Epidemic Models

Having discussed how the Ising Model applies, we will now turn to models inspired by epidemic models. Perhaps the best known example of an epidemic model is the so-called SIR model, which consists of three compartments: Susceptible (individuals who have not yet been affected by the disease), Infected (individuals who currently have the disease), and Removed (individuals who are now immune, either by virtue of having recovered from the disease or by receiving a vaccine).

At this point, our previous finding that interaction networks demonstrate small-world properties is rather convenient, since real social networks also show small-world properties. Note, however, that the average path length appears to be considerably larger in human social networks than in ant colony networks (which makes sense given the relatively confined space of an ant colony). Travers and Milgram famously computed an average path length of 5.2 (giving rise to the famous “six degrees of separation”), as opposed to our previous determination that the average path length for an ant social network is approximately 1.62963.

Travers, J., & Milgram, S. (1969). An Experimental Study of the Small World Problem. *Sociometry*, *32*(4), 425. https://doi.org/10.2307/2786545

In a SI model, susceptible individuals become infected and just remain infected. Its evolution is given by:

where is the spreading rate, is the density of susceptible individuals. and , where and are the absolute number of susceptible and infected people, respectively.

Obviously, if a single vertex in a graph is “infected,” the entire connected component that it’s part of will eventually become infected. If we relax the restrictions on the model somewhat, this could be prevented by promptly quarantining the affected individuals; a notable example of this was Mary Mallon (better known as “Typhoid Mary”), who was the first person in the United States proven to be an asymptomatic carrier of a disease. She was forcibly quarantined after contact tracing found her to be the cause of a large number of cases. She was eventually released on the condition that she not return to her previous occupation of cooking; after she failed to comply with these terms, she was forcibly quarantined for the final two decades of her life.

Wikipedia contributors. (2021, May 21). Mary Mallon. In *Wikipedia, The Free Encyclopedia*. Retrieved 03:35, June 8, 2021, from <https://en.wikipedia.org/w/index.php?title=Mary_Mallon&oldid=1024328309>

A SIS model allows infected individuals to recover; however, no immunity is conferred from having the disease. is the recovery rate.

The model for a SIR is [FROM DYNAMICAL BOOK]:

We will initially start with the SIS model described above. This would perhaps be more aptly named an IAI model (Inactive – Active – Inactive) model.

In this case, we can immediately recognize that and will be the same as and from our previous model – they are simply the rate of departing and arriving foragers. (Video analysis confirms that returning foragers often simply drop their load shortly after return for other ants to bring further down into the nest, which supports our assumption that the rates will, in fact, be the same as the number of arriving and departing foragers). (The Regulation of Ant Colony Foraging Activity without Spatial Information, Gordon). We already calculated that , , and . So,

We will also pick the initial condition

This is, of course, a separable differential equation. We can divide both sides by and multiply both sides by to get:

Which is equivalent to:

We will now integrate both sides:

Rather than work through the full math, we will simply use Matlab’s ODE45 solver to solve for a timespan from 0 to 10:

mu = 0.679592;

k = 62.6;

beta = 0.689091;

tspan = [0 10]

a0 = 0.7

[t, y] = ode45(@(t, a) -(mu \* a) + (beta \* k \* a) - (beta \* k \* a^2), tspan, a0)

plot(t, y, '-o')

This results in the following plot:



The values fairly quickly move from a value of 0.7 to stabilize at a value of approximately 0.9840 (with a small amount of cyclic behavior). This is, of course, rather problematic in that it implies that approximately 98% of foragers are active at all times, which is definitely *not* the case (especially for larger colonies). It also ignores fluctuations – we know that foragers do, in fact, have multiple possible rates. This suggests that something has gone wrong with our model. We need the “stable” baseline rate to be around 70%, and a second “higher” rate.

This does not appear to be a function of the initial value we choose; selecting, for example, or results in a very similar outcome.

However, this does *not* appear to be a function of the model itself; if we select , and , we get the following plot:



In other words, this strongly suggests that the way that the parameters we have chosen and the model interact strongly.

For , , and :



For , , and :



For , , and :



Our problem, then, appears to be twofold: our choice of and our choice of . In fact, since is the total spreading rate, one is initially tempted to simply say that we were mistaken in setting and that we should simply state that

We can simulate this with our original Matlab code simply by changing k = 62.6; to k = 1;

Unfortunately, the situation with this variable doesn’t appear to be much better, as it predicts that the foraging rate will drop from 0.7 to a little over 0.1 within ten time intervals:



Such a conclusion is, of course, roundly contradicted by empirical evidence.

If we continue to utilize , for we get the following plot:



For we get the following:

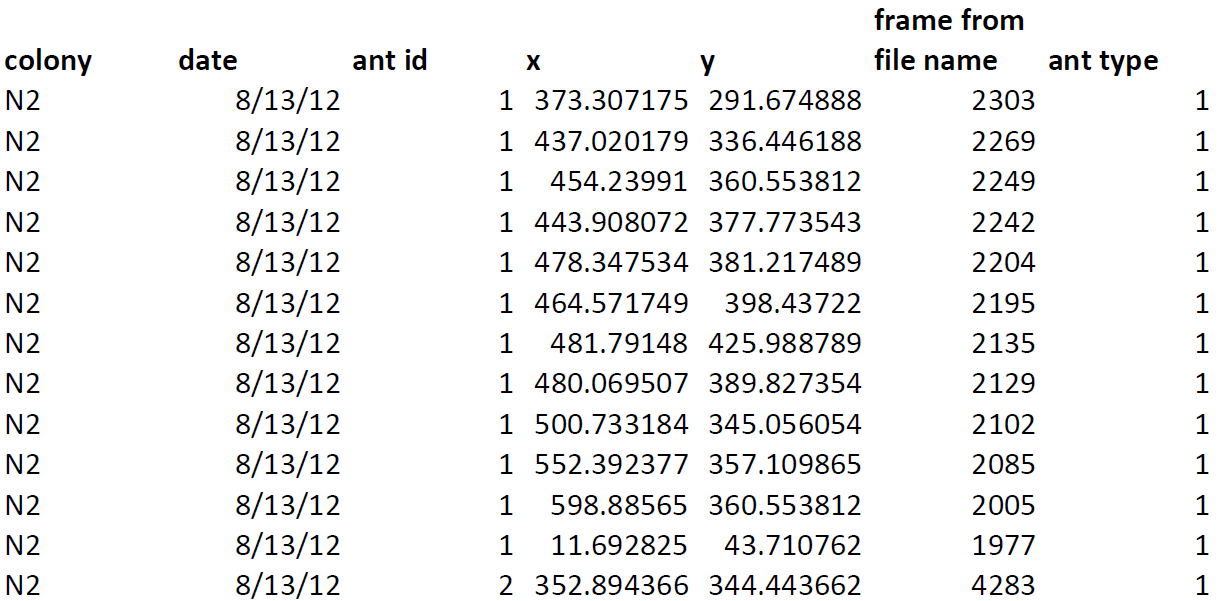


Both are plausible.

We finally arrive at the source of the trouble: our original data set recorded all interactions by forager ants anywhere. This “worked” for our prior models. However, if is the average number of ants that an active forager encounters and is the rate at which successful foragers are returning, then is the rate at which successful foragers are encountering quiescent foragers.

We will use a data set supplied by Pless *et. al.* In this study, the “roof” of an entrance chamber was removed and replaced with a transparent cover and ant interactions were video recorded and later tagged using a Matlab script. (Prior research demonstrated that the light does not substantially alter the ants’ behavior). This data set identifies two types of ants: ants that leave the entrance chamber as active foragers (type 1) and ants that descend into the nest.

An example of the format that the data set is provided in as follows:



Pless, E., Queirolo, J., Pinter-Wollman, N., Crow, S., Allen, K., Mathur, M. B., & Gordon, D. M. (2015). Interactions Increase Forager Availability and Activity in Harvester Ants. *PLOS ONE*, *10*(11). https://doi.org/10.1371/journal.pone.0141971

Conveniently, if we copy this table Adobe Acrobat Reader into Notepad, it preserves the tabular format. By doing a “Replace All” to replace all whitespace with commas, we can easily convert this to a CSV (which will be used below).

In this data set, the ID is not necessarily unique across days or colonies, and the ID is not necessarily chronological by time. The combination of date, colony, and Ant ID uniquely identifies a particular ant. There are two ant types: 1 indicates an outgoing forager, and 2 indicates a descending ant. **X** and **Y** are special coordinates within the chamber. The first entry marks where an ant entered the entrance chamber, the last entry indicates where they left. All other entries represent interactions. So, in the example shown above, the ant enters the chamber at , interacts 11 times, and finally leaves the entrance hall to forage at . Thus, if we group by the colony, date, and ant ID, get the count, and subtract 2 (to exclude entrances and exits), we get the number of interactions:

import pandas as pd

# Read only the columns we need

df = pd.read\_csv(' NestChamberInteractions.csv', usecols = ['colony', 'date', 'antid', 'antType'])

# Filter out all descending ants - we only want ants that become active foragers

df.where(df['antType'] == 1, inplace = True)

# We don't need that column anymore

df.drop(columns = 'antType', inplace = True)

# Add a Count column that's populated with blanks

df['Count'] = [''] \* len(df)

# Get the count and write it to a new CSV file

countDf = df.groupby(['colony', 'date', 'antid']).count()

# Exclude entrances and exits, so that counts represents the true interaction number

countDf['Count'] = countDf['Count'] - 2

countDf.to\_csv('NestChamberInteractionsCounts.csv')

As mentioned in the code comments, we will only examine ants that become active foragers. We have several options for how to get our value of from this. One is to simply look at how many each ant interacts. That will, however, overestimate the number of interactions that take place because the data set does not distinguish between quiescent foragers interacting with successful foragers, successful foragers interacting with other successful foragers, and quiescent foragers interacting with other quiescent foragers. (In fact, some of the ants in the entrance chambers may not be foragers at all but are rather some other class of ant, such as a cleaner).

A second possibility is to estimate how long they are in the entrance chamber on average and compare that to the rate at which foragers are arriving in order to estimate which percent of the interactions are with successful foragers. (We can get their duration in the entrance chamber by calculating the number of video frames that occurred between the first and last entry of a particular ant being in the chamber; since the frame rate is 30 frames per second, we simply divide by 30 to get the number of seconds that they were in the entrance chamber on average).

Another possibility is to determine which pairs of interactions from the same colony and same day have the smallest Euclidean Distance and assume that those are the interactions that take place. (This can be done with a k-d tree, for example, which is part of SciKit-Learn).

We will examine the first two possibilities in turn. First, we will use SAS to perform a distribution analysis of the count data we generated with the code shown above. SAS confirms that this data is, in fact, normally distributed:

| **Goodness-of-Fit Tests for Normal Distribution** | | | | |
| --- | --- | --- | --- | --- |
| **Test** | **Statistic** | | **p Value** | |
| **Kolmogorov-Smirnov** | **D** | 0.12004132 | **Pr > D** | <0.010 |
| **Cramer-von Mises** | **W-Sq** | 0.26724748 | **Pr > W-Sq** | <0.005 |
| **Anderson-Darling** | **A-Sq** | 1.73778206 | **Pr > A-Sq** | <0.005 |



We will now determine how long ants are in the frames. The code is the following:

import pandas as pd

import numpy as np

# Read only the columns we need

df = pd.read\_csv('NestChamberInteractions.csv', usecols = ['colony', 'date', 'antid', 'antType', 'FrameFromFileName'])

# Filter out all descending ants - we only want foragers

df.where(df['antType'] == 1, inplace = True)

# We don't need that column anymore

df.drop(columns = 'antType', inplace = True)

groups = df.groupby(['colony', 'date', 'antid'])

# The first row in a group is the ant entering

entranceFrames = np.array(groups.nth(0).FrameFromFileName)

# The last row in a group is the ant exiting

exitFrames = np.array(groups.nth(1).FrameFromFileName)

frames = np.subtract(entranceFrames, exitFrames)

seconds = np.divide(frames, 30)

pd.DataFrame({'Frames': frames, 'Seconds': seconds}).to\_csv('Ants time in chamber.csv')

We will now run distribution tests on this in SAS:



The skewness and kurtosis suggest that this is not normally distributed. Indeed, , indicating that the data is indeed not normal.

| **Goodness-of-Fit Tests for Normal Distribution** | | | | |
| --- | --- | --- | --- | --- |
| **Test** | **Statistic** | | **p Value** | |
| **Kolmogorov-Smirnov** | **D** | 0.2847429 | **Pr > D** | <0.010 |
| **Cramer-von Mises** | **W-Sq** | 2.6592751 | **Pr > W-Sq** | <0.005 |
| **Anderson-Darling** | **A-Sq** | 13.6566448 | **Pr > A-Sq** | <0.005 |

The unusually low average number of seconds in the entrance chamber is surprisingly low. If foragers are returning at a rate of per second, that would suggest that

This is as opposed to the estimate from the first suggested method.

This has the rather unfortunate effect of 

On the other hand, has the following plot:



Using (based on the median number of interactions) is quite similar.

If we split the difference and set

we end up with the plot:



This appears to be quite satisfactory.

Regardless of the exact value of , then, this model appears to be making reasonable predictions. It also has another extremely important feature: for higher values of , the number of foragers increases (potentially very rapidly). This is exactly consistent with field observations, which indicate that ants very rapidly scale up the number of foragers in response to an increase in the number of successful forager. A key question we have posed previously was whether this behavior is itself adaptive, given that harvester ants are foraging for food that more or less stays in the same place.

Our Ising Model suggested that it’s highly advantageous for the number of active foragers to be based on the probability of success, since this results in lower energy consumption for the colony as a whole; however, the speed at which this change took place is merely a byproduct of local interactions of ants aligning in a way to minimize their individual energy consumption. This model gives further support to this conclusion.