Computational Model of a Vector Mediated Epidemic

Kishen N Gowda (17110074)
Vraj Patel (17110174)
IIT Gandhinagar

Abstract

In this project, we analyze the paper "Computational Model of a Vector Mediated Epidemic" by Adriana Dickman and Ronald Dickman. We study the two methods proposed, the Continuous-time Method and a Discrete-time method. We optimize these two algorithms and provide the running time analysis for both the methods. We found that the parameters used by the authors are too large for us to reproduce and hence we do analysis of some alternative cases.

Contents

1 Introduction

In this project, we have attempted to implement the model and obtain the salient results of the paper "Computational model of a vector-mediated epidemic". The paper is an educational version of a Nature article based on the same topic. The simulation attempts to understand the spreading patterns of vector-mediated diseases. Since epidemic spread is an important social problem, which validates the need for such simulations. Once the mechanism, factors, spread patterns and dynamics of an epidemic are well-understood, measures for controlling the epidemic can be formulated effectively.

Various diseases like malaria and dengue spread through moving infected vectors which infect the host and host infect the uninfected vectors. A simple model proposed by Ross[2] suggests that if the number of vectors are kept in check, the disease should eventually die out. The model failed to take into account the spatial distribution and temporal movement of vectors and location of the hosts with respect to them. Therefore, we will take a look at the model proposed by in the given paper which accounts for these real-life situations.

Though the original authors have provided the explanation in terms of placing the hosts and vectors on a lattice, there is no hard requirement to be restricted to such a simple topology and we therefore provide a simple reformulation of their model in terms of a general graph topology. We will also provide some more implementation level details that should have been provided by the original authors but have failed to do so. In case of one of their methods, we have found a more efficient way of simulation than the one provided. We will also perform a complexity analysis of simulation which would help grasp theoretically the time and computational resources such a simulation requires based on the parameters of the simulation.

2 Modelling a Vector-Borne Disease

The model has two different types of entities, the vectors and the hosts. These entities occupy places on a graph G with N nodes. The vectors are mobile and can hop from one node to other via an edge but the hosts stay on the same node throughout. There is only one host per node. There are two states that each of them can be in: healthy and infected. A healthy vector becomes infected when it bites an infected host. A healthy host becomes infected when an infected vector bites it. The infected vectors may be replaced by a healthy one. An infected host can recover. The number of vectors and the number of hosts remain the same throughout the simulation. In this model, the concept of incubation periods is completely

ignored and it is assumed that each of the entities become infected immediately. We shall now be establishing some notation below:

- The number of hosts is taken as $N_h = N$.
- The number of vectors is taken to be N_v.
- The number of healthy vectors at node i is taken as $v_{0,i}$
- The number of infected vectors at node i will be taken as $v_{1,i}$.
- For each node i, $h_i = 0$ if the host is healthy and $h_i = 1$ if the host is infected.
- A healthy vector will become infect at the rate I_v if it is at the same node as an infected host.
- An infected vector will be replaced with a healthy one with the rate of R_v.
- A vector will move from one node to the other with a rate D. This can be generalised to be D_i for $i \in V(G)$ i.e. different rate of diffusion for each node.
- One infected vector can infect a healthy host at the rate of I_h. Therefore, an uninfected host at node i becomes infected at rate of $v_{1,j} I_h$
- An infected host becomes healthy at the rate of R_h.
- Let $\mathcal{H}$ be number of infected hosts in the system.
- Let $\mathcal{V}$ be the number of infected vectors in the system.

Note that the notation below is based off the original paper [1]. Note that this process involves a change in configuration depending only on the current state and thus is a first order Markov Process.

3 Simulation Algorithms

Now, we discuss the algorithms used to simulate the vector-borne disease described above. The Vector-borne disease problem can be modelled as a Markov process defined in continuous time. Basically, it is defined in terms of the transition rates. The transition rate of the vector-borne disease problem depends on the following transition rates:

1. Recovery of host: For each infected host, this transition is valid, therefore the total rate is $R_H \mathcal{H}$.
2. Replacement of vector: For each infected vector, this transition is valid, therefore the total rate is $R_v \mathcal{V}$.
3. Vector hopping between nodes: Each vector hops at rate D, therefore the total rate is $D \sum_{i \in G} (v_{i,0} + v_{i,1}) = DN_v$.

4. Host Infection: Each uninfected host at node i gets infected at the rate $I_h \nu_{i,0}$. Therefore, the total rate is $I_h \sum_{i \in G} \nu_{i,1}(1 - h_i)$.
5. Vector Infection: Each uninfected vector which is at a node i with an infected host becomes infected at rate I_v. Therefore, the total rate is $I_v \sum_{i \in G} \nu_{0,i} h_i$.

These five transition rates affect the total transition rate of the process.

3.1 Vector-borne disease: Continuous-time algorithm

As this problem is a Markov process defined on continuous time, a simple continuous-time approach should work. As explained above, the total transition rate of the process will be the sum of the transition rates of the five possible events.

$$R = R_h \mathcal{H} + R_v \mathcal{V} + D N_v + I_h \sum_{i \in G} \nu_{i,1}(1 - h_i) + I_v \sum_{i \in G} \nu_{0,i} h_i$$

So, to find the probability of an event to occur, we simply take the ratio of the transition rate of that event to the total transition rate. For example:

$$P_{\text{host recovery}} = \frac{R_h \mathcal{H}}{R}$$

In many cases, we can replace random variables by the average value (Expected value). In this case, this leads to the time step to become equal to $\frac{1}{R}$. Therefore, at every step, we compute R and increment the time by $\frac{1}{R}$.

The original paper mentions about maintaining lists for infected hosts, infected vectors, pairs of healthy hosts and infected vectors occupying the same node, and pairs of infected hosts and healthy vectors occupying the same node. These lists have to be updated on every event. Also, for diffusion, we need an array which stores the current position of each vector [1]. But, this is inefficient and can be optimized to some extent.

Our algorithm maintains array corresponding to the nodes and adjacency list of the graph. On selection of an event, our algorithm does the following:

- **Recovery of Host**: Randomly selects an infected host, based on weights of the hosts, and recovers
- **Replacement of Vector**: Randomly selects an infected vector, based on weights of the vectors, and replaces one infected vector with a healthy vector.
- **Vector Hopping**: Randomly selects a vector, based on weights of the vectors, then uniformly randomly chooses a neighbour node from the adjacency list and transfers this vector to that node.

- **Host Infection**: Randomly select a node, based on weights of infected vectors in that node, and infect that Host.
- **Vector Infection**: Randomly select a node, based on weights of infected host in that node, and infect a vector.

Following is the pseudo code for the algorithm we used:

Algorithm 1 VECTOR-BORNE DISEASE: CONTINUOUS TIME APPROACH

```
1| Initialize the system
2| EVENTS = [Vector Replacement, Host Recovery, Vector Hopping,
            Vector Infection, Host Infection]
3| Choose an event from EVENTS randomly based on weights of the
   events as described above
4| Handle the corresponding event based on the procedure given above
   for each event
5| Increment time t -> t + 1/R
6| Go to step 3
```

Time Complexity: The time complexity of this method can be analyzed to be equal to

$$O(t_{max} \times R_{max} \times N_{reps} \times N)$$

where, $R_{max} = 5 \times max(N_h, N_v) \times max(R_h, R_v, D, I_v, I_h)$

3.2 Vector-borne disease: Discrete-time algorithm

The Continuous-time method was Computationally very heavy based on the model described in the original paper. Our optimizations improves the Computational Complexity to a good extent. But, it turns out that it is still Computationally heavy. The method described above involves considerable expenditure for choosing events and maintaining the data structures (even though well optimized). We now consider a simpler algorithm which involves discrete time. The advantage of such a method is that the entire system gets updated simultaneously in a single pass. Also, the time taken by this single pass is a small but finite duration, denoted by Δt.

In this method, there is no need to maintain a separate list for the vectors or maintain any lists as used in the previous method.

Based on the event selected, the algorithm (as given in [1]) does the following: (n is a binomial random number)

- **Host Recovery/Infection and Vector Infection:** At node j, if the host is infected, then the host recovers with probability $r_h \equiv 1 - exp(-R_h \Delta t)$. And, if there are uninfected vectors at node j, then infect n of them with probability

$$P(n) = \binom{\nu_{0,j}}{n} [1 - exp(-I_\nu \Delta t)]^n exp[-(\nu_{0,j} - n) I_\nu \Delta t]$$

for $n = 0, 1, 2, \ldots, \nu_{0,j}$. If the host at site j is not affected, then the host becomes infected with a probability $1 - exp(-\nu_{1,j} I_h \Delta t)$

- **Vector Replacement**: At site j, n of the infected vectors are replaced by uninfected vectors, with probability

$$P(n) = \binom{\nu_{1,j}}{n} [1 - exp(-R_\nu \Delta t)]^n exp[-(\nu_{1,j} - n) R_\nu \Delta t]$$

for $n = 0, 1, 2, \ldots, \nu_{1,j}$.

- **Vector Hopping**: At site j, n vectors hop, with probability

$$P(n) = \binom{\nu_{1,j}}{n} [1 - exp(-D \Delta t)]^n exp[-(\nu_{1,j} - n) D \Delta t]$$

for $n = 0, 1, 2, \ldots, \nu_{1,j}$. Choose the new site for each hopping vector from its set of neighbours using the adjacency list.

The binomial probability distributions associated with the vector infection, replacement and hopping are stored in lookup tables. [1] In Python, lru_cache from functools can be used for this purpose.

Time Complexity: On proper analysis, we can get the time complexity of the discrete-time method as

$$O\left(\left\lceil \frac{t_{max}}{\Delta t} \right\rceil \times N_{rep} \times N\right)$$

4 Results

Due to the computational complexity of the operations as explained in the previous sections, we could not run the simulations for finding critical parameter as stated in the paper. We will work with smaller, less computationally intense parameters and vary them to qualitatively analyse the results that are obtain and how they tie in to the real-world interpretation of the model. We define the vector density (ρ_ν) as the average number of vectors per node. Also, N_{rep} is the no. of times you have to repeat the experiment.

4.1 Continuous-time Method

The parameters used are: $\mathrm{N} = 100, \mathrm{t_{max}} = 180, \mathrm{N_{rep}} = 10000, \mathrm{R_h} = 2.0, \mathrm{R_v} = 1.5, \mathrm{I_h} = 1.0, \mathrm{I_v} = 2.0, \mathrm{D} = 0.5$

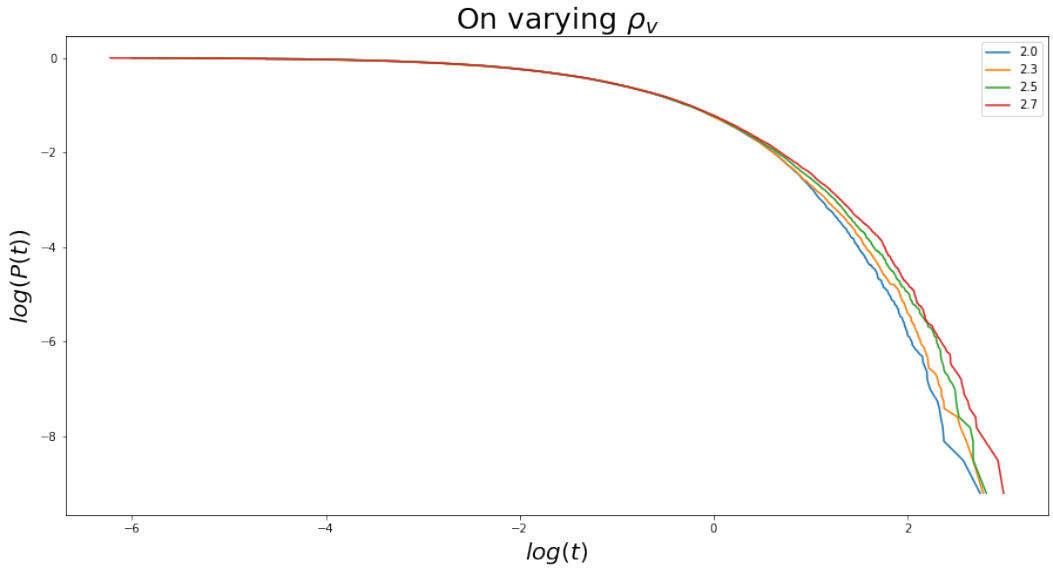

Figure 1 Log-log plot of P(t) vs t

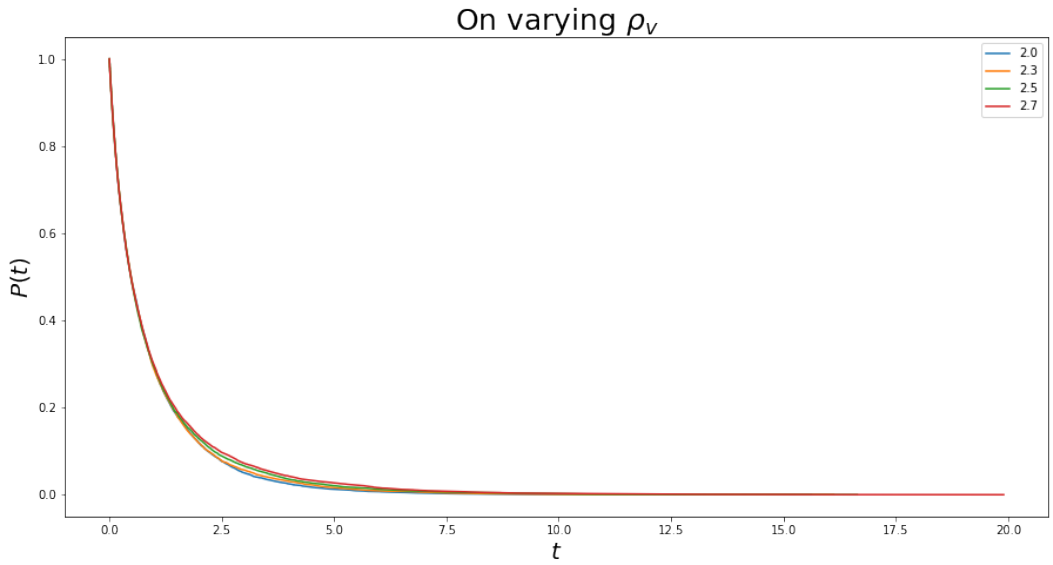

Figure 2 Plot of P(t) vs t

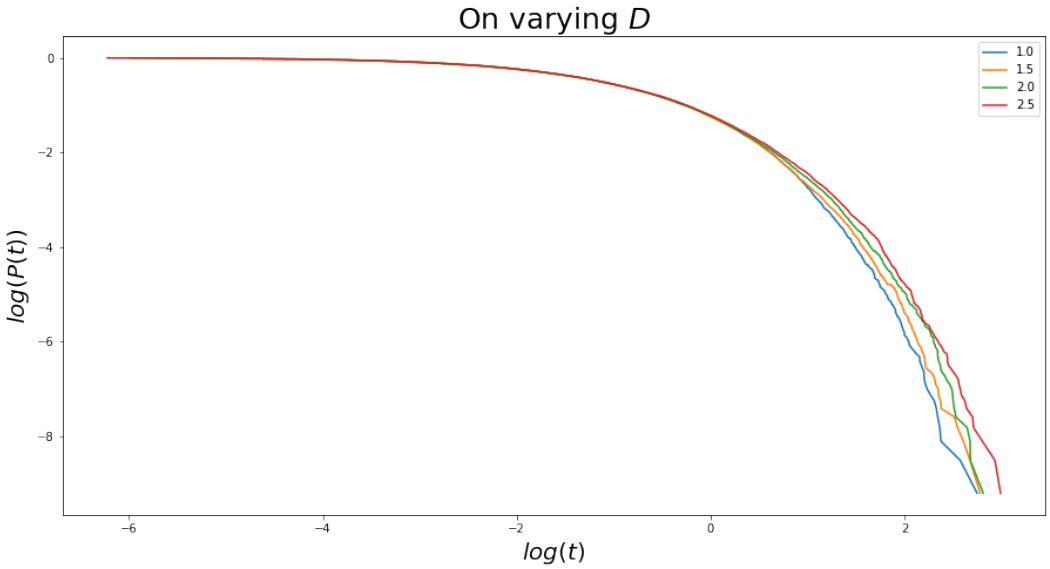

Figure 3 Log-log plot of P(t) vs t on varying D

4.2 Discrete-time Method

The parameters used are: $N = 5000, t_{max} = 4000, N_{rep} = 1000, R_h = 1.0, R_v = 1.5, I_h = 1.0, I_v = 2.0, D = 0.5, \Delta t = 0.1$

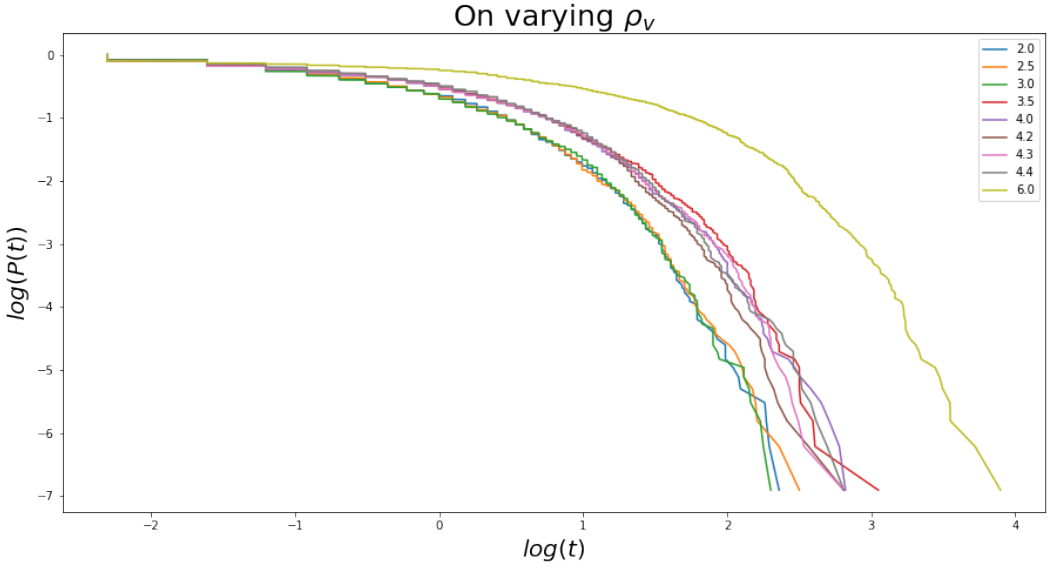

Figure 4 Log-log plot of P(t) vs t

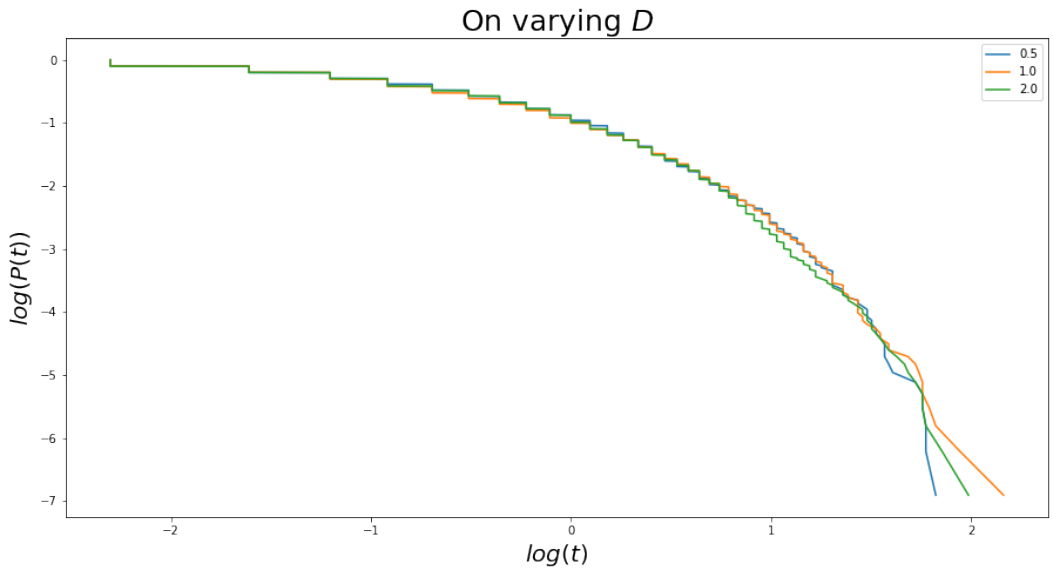

Figure 5 Log-log plot of P(t) vs t on varying D

5 Conclusion

We observe that on increasing the vector density (ρ_v) and rate of diffusion D keeping everything fixed, you get an increased probability of the epidemic persisting for a longer time. We could not compare our plots with the ones given by the original authors because of the lack of computation power despite making good optimizations to the code and algorithm. In this work, we have discussed the two methods proposed by *Dickman et. al.*[1] to simulate the spread of Vector-borne epidemic, a continuous time approach and a discrete time approach. In Statistical Mechanics, this type of model is of interest as it exhibits a phase transition. Moreover, the two methods are simple and intuitive. The continuous time model is more faithful towards the original Markov process, but the only problem is that it is computationally expensive. The discrete time approach improves in terms of computational complexity.

6 References

[1] Adriana Dickman and Ronald Dickman. 'Computational model of a vector-mediated epidemic'. In: *American Journal of Physics* 83 (May 2015), pp. 468–474. DOI: 10.1119/1.4917164.

[2] Ronald Ross. 'An Application of the Theory of Probabilities to the Study of a priori Pathometry. Part I'. In: *Proceedings of the Royal Society of London. Series A, Containing*

Papers of a Mathematical and Physical Character 92.638 (1916), pp. 204–230. ISSN: 09501207. URL: http://www.jstor.org/stable/93760.