Computational Model of a Vector Mediated Epidemic

PH599 Computational Physics

Vraj Patel | Kishen N. Gowda

November 24, 2019

Indian Institute of Technology Gandhinagar

Introduction

Introduction

- Based on the paper ”Computational Model of a Vector Mediated Epidemic”[1]

Introduction

- Based on the paper ”Computational Model of a Vector Mediated Epidemic”[1]
- The paper attempts to understand the spreading patterns of vector mediated diseases like Malaria, Dengue, etc.

Introduction

- Based on the paper **”Computational Model of a Vector Mediated Epidemic”**[1]
- The paper attempts to understand the spreading patterns of vector mediated diseases like Malaria, Dengue, etc.
- Epidemic spread is an important Social problem.

Introduction

- Based on the paper **”Computational Model of a Vector Mediated Epidemic”**[1]
- The paper attempts to understand the spreading patterns of vector mediated diseases like Malaria, Dengue, etc.
- Epidemic spread is an important Social problem.
- Understanding the spreading pattern helps in formulating measures to control it.

Introduction

- Diseases like Malaria, Dengue, etc. can be modelled as a vector-borne disease

Introduction

- Diseases like Malaria, Dengue, etc. can be modelled as a vector-borne disease
- Spread by infected vectors infect hosts and infected hosts infect uninfected vectors.

Introduction

- Diseases like Malaria, Dengue, etc. can be modelled as a vector-borne disease
- Spread by infected vectors infect hosts and infected hosts infect uninfected vectors.
- Here, Mosquito $\implies$ Vector, Human $\implies$ Host

Modelling a Vector-Borne Disease

Host-Vector Model

- Two entities: Vector and Host

Host-Vector Model

- Two entities: Vector and Host
- Entities occupy the nodes of a Graph (N nodes)

Host-Vector Model

- Two entities: Vector and Host
- Entities occupy the nodes of a Graph (N nodes)
- Vectors are mobile, can hop to neighbour nodes

Host-Vector Model

- Two entities: Vector and Host
- Entities occupy the nodes of a Graph (N nodes)
- Vectors are mobile, can hop to neighbour nodes
- Hosts are stationary

Host-Vector Model

- Two entities: Vector and Host
- Entities occupy the nodes of a Graph (N nodes)
- Vectors are mobile, can hop to neighbour nodes
- Hosts are stationary
- Only one host per node

Host-Vector Model

- Two entities: Vector and Host
- Entities occupy the nodes of a Graph (N nodes)
- Vectors are mobile, can hop to neighbour nodes
- Hosts are stationary
- Only one host per node
- Hosts and vectors can either be infected or healthy

Host-Vector Model

- Healthy vector $\implies$ bites infected host $\implies$ become infected

Host-Vector Model

- Healthy vector $\implies$ bites infected host $\implies$ become infected
- Infected vector $\implies$ bites healthy host $\implies$ host becomes infected

Host-Vector Model

- Healthy vector $\implies$ bites infected host $\implies$ become infected
- Infected vector $\implies$ bites healthy host $\implies$ host becomes infected
- Infected host recovers

Host-Vector Model

- Healthy vector $\implies$ bites infected host $\implies$ become infected
- Infected vector $\implies$ bites healthy host $\implies$ host becomes infected
- Infected host recovers
- Infected vector can be replaced by a healthy vector

Host-Vector Model

- Healthy vector $\implies$ bites infected host $\implies$ become infected
- Infected vector $\implies$ bites healthy host $\implies$ host becomes infected
- Infected host recovers
- Infected vector can be replaced by a healthy vector
- No. of vectors and No. of hosts remain constant throughout

Notation

- No. of hosts $N_h = N$.

Notation

- No. of hosts $N_h = N$.
- No. of vectors N_v.

Notation

- No. of hosts $N_h = N$.
- No. of vectors N_v.
- No. of healthy vectors at node i = $v_{0,i}$

Notation

- No. of hosts $N_h = N$.
- No. of vectors N_v.
- No. of healthy vectors at node i = $v_{0,i}$
- No. of infected vectors at node i = $v_{1,i}$.

Notation

- No. of hosts $N_h = N$.
- No. of vectors N_v.
- No. of healthy vectors at node i = $v_{0,i}$
- No. of infected vectors at node i = $v_{1,i}$.
- Health of host, h_i. 0 if healthy and 1 if infected.

Notation

- No. of hosts $N_h = N$.
- No. of vectors N_v.
- No. of healthy vectors at node i = $v_{0,i}$
- No. of infected vectors at node i = $v_{1,i}$.
- Health of host, h_i. 0 if healthy and 1 if infected.
- Vector infection rate I_v, if present at node with infected host.

Notation

- Vector replacement rate R_v.

Notation

- Vector replacement rate R_v.
- Vector Hopping rate D.

Notation

- Vector replacement rate R_v.
- Vector Hopping rate D.
- Host infection rate I_h. Uninfected host at node i becomes infected at rate $v_{1,j}I_h$

Notation

- Vector replacement rate R_v.
- Vector Hopping rate D.
- Host infection rate I_h. Uninfected host at node i becomes infected at rate $v_{1,j} I_h$
- Host recovery rate R_h.

Notation

- Vector replacement rate R_v.
- Vector Hopping rate D.
- Host infection rate I_h. Uninfected host at node i becomes infected at rate $v_{1,j} I_h$
- Host recovery rate R_h.
- No. of infected hosts in the system = $\mathcal{H}$.

Notation

- Vector replacement rate R_v.
- Vector Hopping rate D.
- Host infection rate I_h. Uninfected host at node i becomes infected at rate $v_{1,j}I_h$
- Host recovery rate R_h.
- No. of infected hosts in the system = $\mathcal{H}$.
- No. of infected vectors in the system = $\mathcal{V}$.

Simulation Algorithms

Simulation Algorithms

Transition rate of the system depends on the following transition rates:

- **Recovery of host**: For each infected host, this transition is valid, therefore the total rate is $R_H\mathcal{H}$.

Simulation Algorithms

Transition rate of the system depends on the following transition rates:

- **Recovery of host**: For each infected host, this transition is valid, therefore the total rate is $R_H \mathcal{H}$.
- **Replacement of vector**: For each infected vector, this transition is valid, therefore the total rate is $R_v \mathcal{V}$.

Simulation Algorithms

Transition rate of the system depends on the following transition rates:

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

Simulation Algorithms

Transition rate of the system depends on the following transition rates:

- **Host Infection**: Each uninfected host at node i gets infected at the rate $I_h v_{i,0}$. Therefore, the total rate is $I_h \sum_{i \in G} v_{i,1}(1 - h_i)$.

Simulation Algorithms

Transition rate of the system depends on the following transition rates:

- **Host Infection**: Each uninfected host at node i gets infected at the rate $I_h v_{i,0}$. Therefore, the total rate is $I_h \sum_{i \in G} v_{i,1}(1 - h_i)$.
- **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 $I_v \sum_{i \in G} v_{0,i} h_i$.

Continuous Time Approach

- Total transition rate = Sum of transition rates of different events

Continuous Time Approach

- Total transition rate = Sum of transition rates of different events
- $R = R_h\mathcal{H} + R_v\mathcal{V} + DN_v + I_h \sum_{i \in G} v_{i,1}(1 - h_i) + I_v \sum_{i \in G} v_{0,i}h_i$

Continuous Time Approach

- Total transition rate = Sum of transition rates of different events
- $R = R_h\mathcal{H} + R_v\mathcal{V} + DN_v + I_h \sum_{i \in G} v_{i,1}(1 - h_i) + I_v \sum_{i \in G} v_{0,i} h_i$
- For example, $P_{host\ recovery} = \frac{R_h\mathcal{H}}{R}$

Continuous Time Approach

- Total transition rate = Sum of transition rates of different events
- $R = R_h \mathcal{H} + R_v \mathcal{V} + DN_v + I_h \sum_{i \in G} v_{i,1}(1 - h_i) + I_v \sum_{i \in G} v_{0,i} h_i$
- For example, $P_{host\ recovery} = \frac{R_h \mathcal{H}}{R}$
- $p(dt) = Re^{-Rdt}$. But, we can take $dt = \frac{1}{R}$, the average value. Improves performance significantly.

Continuous Time Approach: Events

- **Recovery of Host**: Randomly selects an infected host, based on weights of the hosts, and recovers

Continuous Time Approach: Events

- 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.

Continuous Time Approach: Events

- **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.

Continuous Time Approach: Events

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

Continuous Time Approach: Events

- **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.

Continuous Time Approach: Pseudo Code

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
```

Continuous Time Approach: Time Complexity

$$O\left(t_{max} \times R_{max} \times N_{rep} \times N\right)$$

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

$N_{rep} =$ no. of times the experiment is repeated.

Discrete Time Approach

- Involves discrete time

Discrete Time Approach

- Involves discrete time
- Advantage: The entire system gets updated simultaneously in a single pass.

Discrete Time Approach

- Involves discrete time
- Advantage: The entire system gets updated simultaneously in a single pass.
- Time taken by this single pass is a small but finite duration, denoted by Δt.

Discrete Time Approach

- Involves discrete time
- Advantage: The entire system gets updated simultaneously in a single pass.
- Time taken by this single pass is a small but finite duration, denoted by Δt.
- In this method, no need to maintain a separate list for the vectors.

Discrete Time Approach: Events

- **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

$$P(n) = \binom{v_{0,j}}{n} (1 - e^{-I_v \Delta t})^n (e^{-I_v \Delta t})^{(v_{0,j} - n)}$$

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

Discrete Time Approach: Events

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

$$P(n) = \binom{v_{1,j}}{n}(1 - e^{-R_v \Delta t})^n (e^{-R_v \Delta t})^{(v_{1,j} - n)}$$

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

Discrete Time Approach: Events

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

$$P(n) = \binom{v_{1,j}}{n}(1 - e^{-D\Delta t})^{n}(e^{-D\Delta t})^{(v_{1,j}-n)}$$

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

Discrete Time Approach: Pseudo Code

Vector-borne disease: Discrete time approach

```
1| Initialize the system
2| EVENTS = [Vector Replacement, Host Recovery, Vector Hopping,
            Vector Infection, Host Infection]
3| Perform all the events as explained above.
4| Store the values
5| Go to step 3
```

Discrete Time Approach: Time Complexity

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

where, N_{rep} =no. of times the experiment is repeated.

Results

Results

- High computational complexity of the operations as explained in the previous sections.

Results

- High 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.

Results

- High 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.

Results

- High 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.
- We define the vector density (ρ_V) as the average number of vectors per node.

Continuous Time Method: Parameters

Parameters:

- $N = 100$
- $t_{max} = 180$
- $N_{rep} = 10000$
- $R_h = 2.0$
- $R_v = 1.5$
- $I_h = 1.0$
- $I_v = 2.0$
- $D = 0.5$

Continuous Time Method: Plots

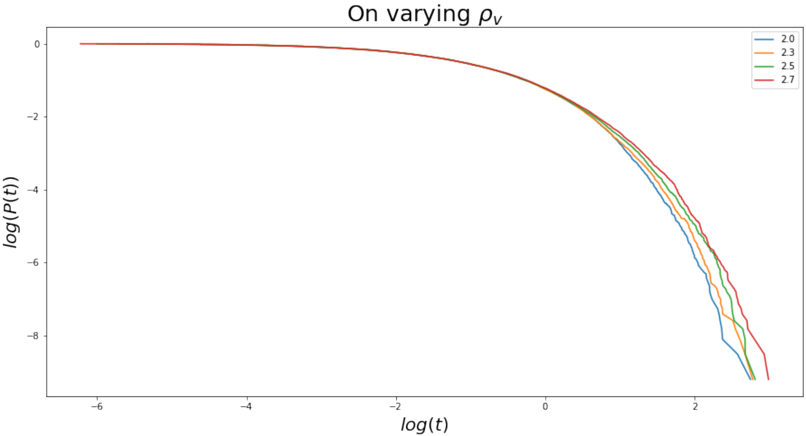

Figure 1: Log-log plot of $P(t)$ vs t

Continuous Time Method: Plots

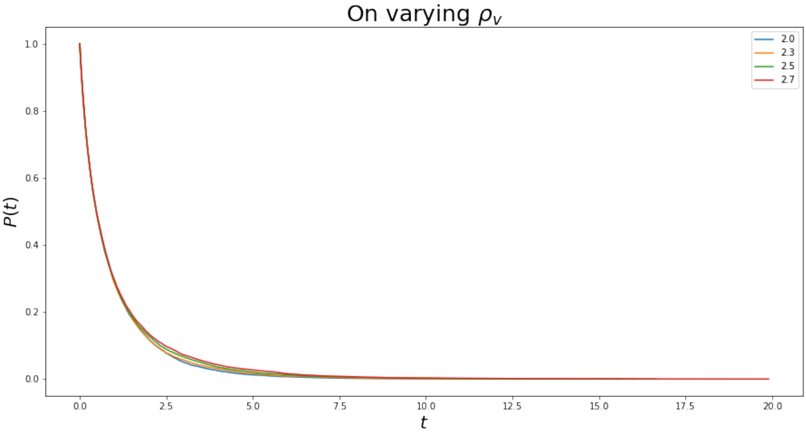

Figure 2: Plot of $P(t)$ vs t

Continuous Time Method: Plots

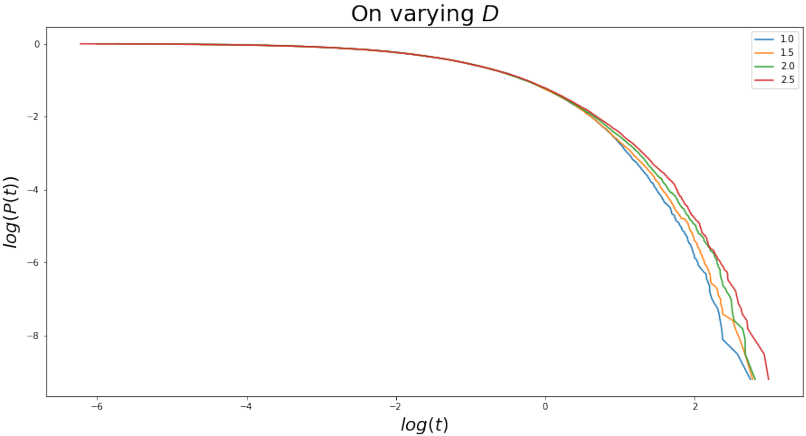

Figure 3: Log-log plot of $P(t)$ vs t on varying D

Discrete Time Method: Parameters

Parameters:

- $N = 5000$
- $t_{max} = 4000$
- $N_{rep} = 1000$
- $R_h = 1.0$
- $R_v = 1.5$
- $I_h = 1.0$
- $I_v = 2.0$
- $D = 0.5$
- $\Delta t = 0.1$

Discrete Time Method: Plots

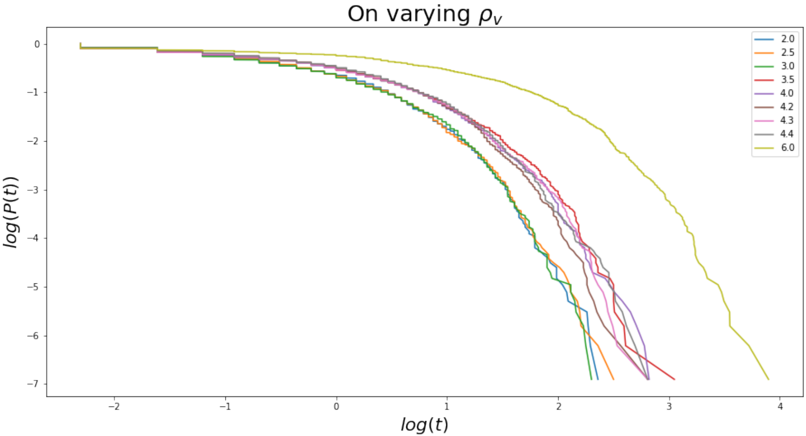

Figure 4: Log-log plot of $P(t)$ vs t

Discrete Time Method: Plots

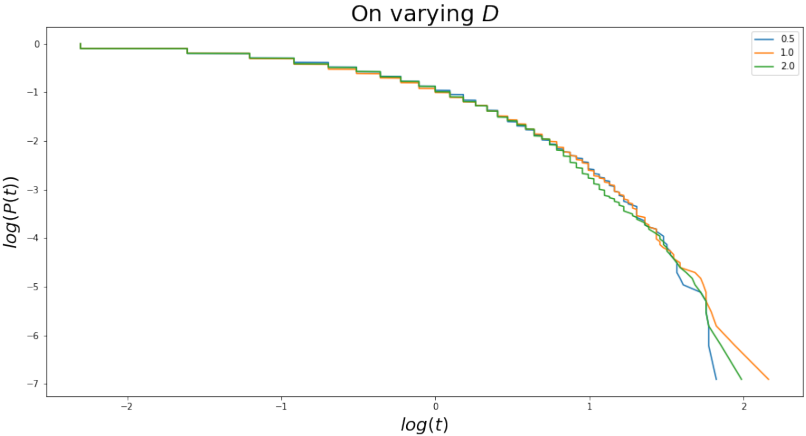

Figure 5: Log-log plot of $P(t)$ vs t on varying D

References

Dickman, A., and Dickman, R.
Computational model of a vector-mediated epidemic.
American Journal of Physics 83 (05 2015), 468–474.

Questions?