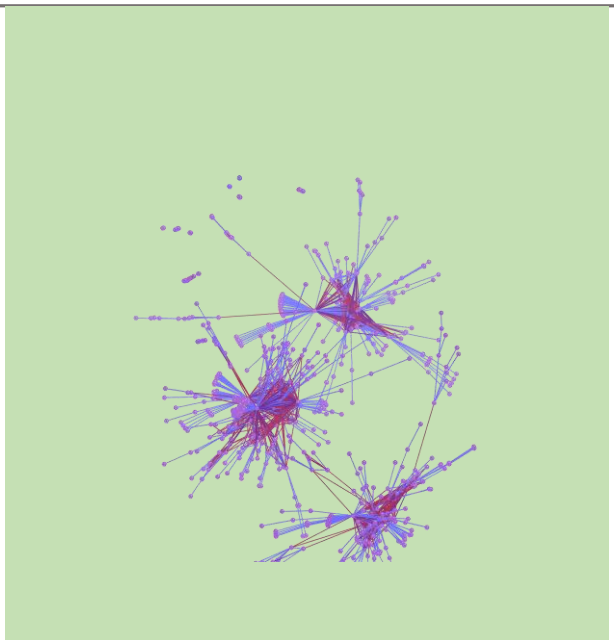

GEMF_Tool

Epidemic Model Simulation Software Tool

Faryad Darabi Sahneh and Caterina Scoglio
Electrical and Computer Eng. Department
Kansas State University

Contents

GEMF_Tool: Simulation Software Tool for Spreading Processes in Multilayer Networks

Introduction

This software tool implements the generalized formulation of the epidemic spreading problem and the related modeling solution proposed by Sahneh et al. in [1]. It considers a spreading process among N nodes which can be in M different compartments, and where nodes interact through links of multiple types, forming a multi-layer network.

The modeling starts with a simple node level description of the underlying stochastic process, where nodes can transition through different states following specific transition rules. The model provides exact equations of the Markov process, which describe the time evolution of the state occupancy probabilities, and a mean-field type approximation for the same occupancy probabilities [1].

GEMF has a simple structure, being characterized by the Laplacian of the transition rate graphs and the elements of the adjacency matrices of the network layers. This simple and general structure, explained in page 5, provides a direct path for a systematic procedure to simulate all network-based epidemic models which fall into the above general description.

In the next sections, we first describe 1) the theoretical foundation of GEMF_Tool, 2) several examples of epidemic models that can be simulated, and 3) the difference between stochastic and mean-field solutions. Second, we list the software modules and the functions that constitute GEMF_Tool, and how to input networks and model's parameters and how to collect output results. Finally, we show how to use GEMF_Tool for the examples of epidemic models described in the first section.

What are the theoretical foundations of this software?

GEMF_Tool is based on the mathematical description in [1]. In the following, we shortly summarize the main characteristics of those foundations. First, we explain how a generic network topology is considered in the spreading process. Then, we describe the exact Markov process and the mean-field approximation, both approaches providing spatio-temporal evolutions of an epidemic.

Epidemics on Networks

Consider a network of N nodes where the contact is determined by the adjacency matrix A. Node j is a neighbor of i, if it can transmit the infection to node i through link (i, j). If j is a neighbor of i, then $a_{ij} = 1$; otherwise, $a_{ij} = 0$. We assume that the collective system is a Markov process. In the simplest epidemic model — the susceptible-infected-susceptible (SIS) model — the state $x_i(t)$ of a node i at time t is a Bernoulli random variable, where $x_i(t) = 0$ if node i is susceptible and $x_i(t) = 1$ if it is infected. The expected value of $x_i(t)$ is the infection probability of node i. The recovery process for infected agent i has an exponential time distribution with average duration $1/\delta$, where $\delta \in R^+$ is called the recovery rate. Similarly, the infection process for susceptible node i in contact with infected node $j \neq i$ has an exponential time distribution with average duration $1/\beta$, where $\beta \in R^+$ is called the infection rate. Therefore, a susceptible node effectively becomes infected at rate $\beta Y_i(t)$, where $Y_i(t)$ is the number of infected neighbors of agent i at time t,

$$Y_i(t) = \sum_{j=1}^{N} a_{ij} x_j(t)$$

Based on this simple transition mechanism, Figure 1 shows how the state of a generic node is governed by a simple transition graph. The neighbor's state that causes transitions from susceptible to infected states is the infected state I; we call this the *inducer* state. The transition from S to I is given by a triad that includes the inducer state I, the infection rate β, and the network topology identifying neighbors, the contact network N. The variable N is used here to identify both the contact network and the number of its nodes. Overall, the transition is governed by (I, β, N). This type of transition is called *edge-based transition*. The transition from infected state to susceptible is only governed by the recovery rate δ, and does not depend on the node's neighbors, nor is provoked by an inducer compartment. This type of transition is called *nodal transition*.

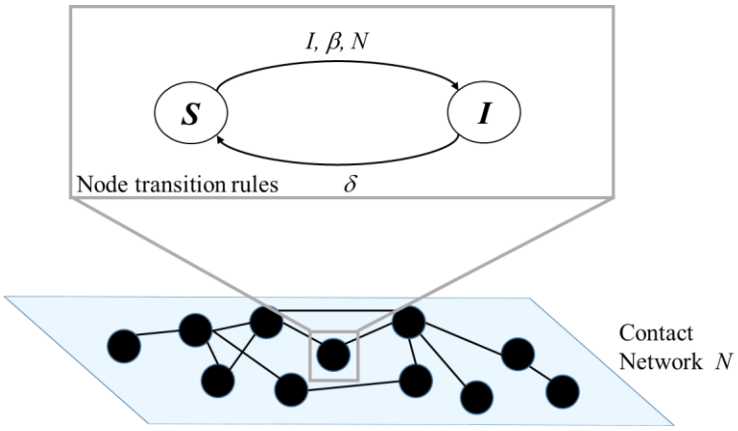

Figure 1. Schematic of the network-based SIS model

It is clear as this model do not assume any network model, nor it aggregates characteristics of the network. The network topology is fully considered through its complete adjacency matrix, showing why GEMF is a model that allows the simulation of spreading processes on *any possible contact network*.

The SIS model is a good example of how a simple compartmental model at the node level along with a network topology can lead to very rich and complex dynamics. Following the structure and underlying assumptions of existing epidemic models, we proposed a generalized individual-based spreading model where 1) the node model can have multiple compartments, and 2) the network topology can have multiple layers. Both generalizations are very important, and provide the theoretical foundations for the tremendous flexibility of GEMF_Tool.

Generalization of Markov Process Approach

One of the generalizations of GEMF concerns the compartment set, where each node can be in one compartment in the set $S = \{s_1, s_2, .., s_m, .., s_M\}$. For example, in the SIS model for epidemic spread, $M = 2$ and S = {Susceptible, Infected}. Each compartment is labeled with a number from 1 to M. The node state $x_i(t)$ of node i at time t is $x_i(t) = e_m$ if node i is in compartment m at time t. Here, e_m is the m-th standard unit vector in the R^M Euclidean space, i.e., all entries of e_m are zero except for the m-th entry, which is equal to one. Therefore, the expected value of $x_i(t)$ is in fact the compartment occupancy probability vector, i.e.,

$$E[x_i] = [\Pr[x_i = e_1], ..., \Pr[x_i = e_M]]^T$$

The state of a single node is not enough to describe the evolution of the node state. Instead, the joint state of all nodes follows a Markov process. Therefore, the network state at time t, denoted by $X(t)$, is the joint state of all nodes defined as:

$$X(t) = \otimes_{i=1}^{N} x_i(t) = x_1(t) \otimes x_2(t) \otimes ... \otimes x_N(t)$$

where $\otimes$ is the Kronecker product.

The other generalization in GEMF concerns the topology. In the SIS model, the interaction among nodes is represented by the contact network. However, the types of interaction can be different in a complex network. To provide flexibility, we allow the topology $G = (V, E_1, E_{2,...,} E_L)$ to be consisting of L layers of graphs where V is the set of nodes, and E_l is the set of edges that represent the interaction between each pair of individuals in the l-th layer. These graphs have the same nodes, but the edges can be different. Figure 2 exemplify a two-layer network, where nodes are repeated for clarity in both layers.

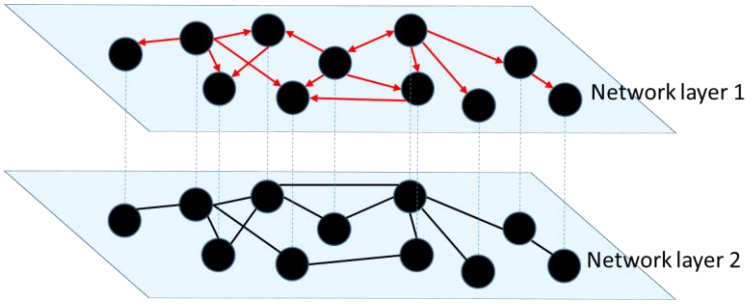

Figure 2. Example of a two layer network. Nodes are the same in both layers, while links are different

With no loss of generality, we assume each layer corresponds to one and only one inducer compartment.

The node level description of the Markov process can be expressed as follows:

$$\Pr[x_i(t+\Delta t) = e_n \mid x_i(t) = e_m, X(t)] = \delta_{mn}\Delta t + \Delta t \sum_{l=1}^{L} \beta_{l,mn} y_{l,i}(t) + o(\Delta t), \text{ for } i = \{1, ..., N\} \text{ and } m \neq n,$$

where $y_{l,i}(t) \triangleq \sum_{j=1}^{N} a_{l,ij} 1_{\{x_j(t)=e_{q_l}\}}$ is the number of neighbors of nodes *i* in G_l that are in the corresponding inductor compartment q_l, $\beta_{l,mn}$ is the edge-based transition rate from compartment *m* to compartment *n* for layer *l*, and δ_{mn} is the nodal transition from compartment *m* to *n*, which does not depend on the neighbors in any network layer.

The Equation above provides a useful node-level description of the Markov process, and it is used in GEMF_Tool for Monte Carlo numerical simulations of the spreading process. See [1] for a detailed description of the equation derivation.

Mean-Field Approximation

Using first order mean-field type approximation, the joint expected values are approximated in terms of marginal expected values, and the time evolution of the expected values is described through a set of ordinary differential equations with MN states [1].

Denoting by $v_i(t)$ the expected value of x_i at time t, i.e., $v_i(t) \triangleq E[x_i(t)]$, a new set of differential equations is obtain to describe the state evolution of each node specifying our *generalized epidemic mean-field model GEMF*:

$$\frac{dv_i}{dt} = -Q_\delta^T v_i - \sum_{l=1}^{L} \left(\sum_{j=1}^{N} a_{l,ij} v_{j,q_l}\right) Q_{\beta_l}^T v_i, \qquad i = \{1,\ldots,N\},$$

where the generalized transition matrices $Q_\delta \in \mathbb{R}^{M\times M}$ and $Q_{\beta_l} \in \mathbb{R}^{M\times M}$ are defined as

$$(Q_\delta)_{mn} \triangleq -\delta_{mn}, \left(Q_{\beta_l}\right)_{mn} \triangleq -\beta_{l,mn}, m \neq n$$
$$(Q_\delta)_{mm} \triangleq \sum_{n\neq m} \delta_{mn}, \left(Q_{\beta_l}\right)_{mn} \triangleq \sum_{n\neq m} \beta_{l,mn}.$$

Matrices Q_δ and Q_{β_l} are actually the Laplacian matrices of transition rate graphs. In Figure 3, transition rate graphs for the SIS model are shown.

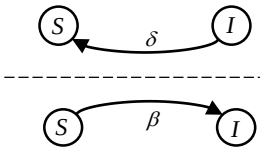

Figure 3. Transition rate graphs of SIS

We recommend a careful study of paper [1] for a thorough and complete understanding of the above derivations.

What type of processes can I simulate?

The SIS model gives very good insights into how to properly define the transition possibilities to describe an epidemic spreading process. In the SIS model, there are two transitions. The transition from the *infected* state to the *susceptible* represents the recovery process, and occurs independently of the states of other nodes. We call this type of transition as *node-based transition*. The transition from the *susceptible* state to

the *infected* state represents the infection process, and occurs as a function of the number of infected neighbors. We call this type of transition as *edge-based transition*. This transition depends on the network topology. When creating an epidemic model, the *node transition graph* needs to be defined, to systematically list (1) states, (2) transitions and (3) network layers

1. Epidemic states or compartments

When defining an epidemic model, the first decision is the set of compartments. The SIS has only two compartments, namely susceptible and infected; SIR has three compartments, namely susceptible, infected, and recovered; the SEIR model has four compartments, namely susceptible, exposed, infected, and recovered.

2. Transitions, type, inducer compartments, and network layers

The second decision is about how to transition from one compartment to another. To this end, in addition to define a transition rate parameter, also the nature of the transition needs to be specified; either node-based or edge-based. In the case of an edge-based transition, the inducer compartment and the layer that defines the neighbor nodes have to be specified too.

Examples

We now give several examples of the above process to define the *node transition graph.* Keep in mind that these *node transition graphs* only represent the transition mechanism for each node in the network and not for the entire population.

SIS on a single-layer network

In the Susceptible-Infected-Susceptible (SIS) model, each node can be either *susceptible* or *infected*. Hence, the number of compartments, denoted by *M*, in the SIS model is *M* = 2. A susceptible node can become infected if it is surrounded by infected nodes. The infection process of a node with one infected neighbor is a Poisson process with transition rate β. The infection processes are stochastically independent of each other. Therefore, for a susceptible node with more than one infected node in its neighborhood, the transition rate is the infection rate β times the number of the infected neighbor nodes. The neighborhood of each node is determined by a contact network N. In addition to the infection process, there exists also a recovery process. An infected node returns to be susceptible with a curing rate δ. A table of the main characteristics and a schematic for the SIS model are shown in the following Table 1 and Figure 4.

Table 1. Descriptors of the SIS model

Susceptible-Infected-Susceptible					
States	Transition	Type	Transition parameter	Inducer	Layer
S (susceptible) I (infected)	$(S \to I)$	edge-based	β	Neighbors in state I	Contact Network N
	$(I \to S)$	node-based	δ		

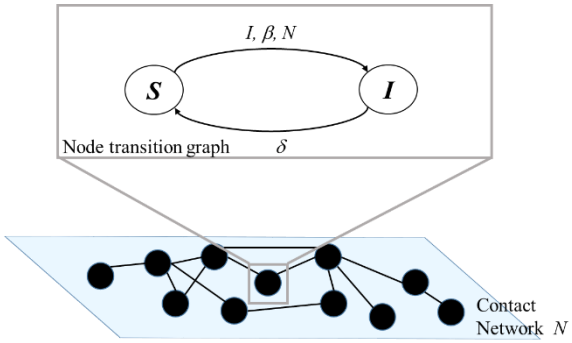

Figure 4. Node transition graph for the SIS model

SIR on a single-layer network

In the Susceptible-Infected-Recovered (SIR) model, each node can be either *susceptible, infected,* or *recovered* and *immune*. Hence, the number of compartments, denoted by *M*, in the SIR model is *M* = 3. A susceptible node can become infected if it is surrounded by infected nodes. The infection process of a node with one infected neighbor is a Poisson process with transition rate β. The infection processes are stochastically independent of each other. Therefore, for a susceptible node with more than one infected node in its neighborhood, the transition rate is the infection rate β times the number of the infected neighbor nodes. The neighborhood of each node is determined by a contact network *N*. In addition to the infection process, there exists also a recovery process. An infected node becomes recovered and immune with a recovery rate δ. A table of the main characteristics and a schematic for the SIR model are shown in the following Table 2 and Figure 5

Table 2. Descriptors of the SIR model

Susceptible-Infected-Recovered					
States	Transition	Type	Transition parameter	Inducer	Layer
S (susceptible) I (infected) R (recovered)	$(S \rightarrow I)$	edge-based	β	Neighbors in state I	Contact Network N
	$(I \rightarrow R)$	node-based	δ		

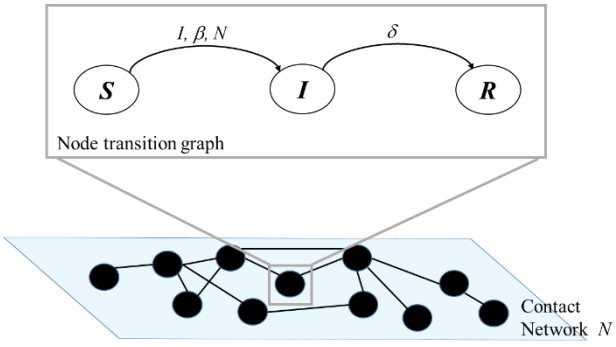

Figure 5. Node transition graph for the SIR model

SEIR on a single-layer network

In the Susceptible-Exposed-Infected-Recovered (SEIR) model, each node can be either *susceptible, exposed*, *infected,* or *recovered* and *immune*. Hence, the number of compartments, denoted by *M*, in the SEIR model is *M* = 4. A susceptible node can become exposed if it is surrounded by infected nodes. The infection process of a node with one infected neighbor is a Poisson process with transition rate β. The infection processes are stochastically independent of each other. Therefore, for a susceptible node with more than one infected node in its neighborhood, the transition rate is the infection rate β times the number of the infected neighbor nodes. The neighborhood of each node is determined by a contact network *N*. An exposed node is not yet infectious, but it will transition to the infected state with rate λ. Finally, an infected node becomes recovered and immune with a recovery rate δ. A table of the main characteristics and a schematic for the SEIR model are shown in the following Table 3 and Figure 6.

Table 3. Descriptors of the SEIR model

Susceptible-Exposed-Infected-Recovered					
States	Transition	Type	Transition parameter	Inducer	Layer
S (susceptible) E (exposed) I (infected) R (recovered)	$(S \rightarrow E)$	edge-based	β	Neighbors in state I	Contact Network N
	$(E \rightarrow I)$	node-based	λ		
	$(I \rightarrow R)$	node-based	δ		

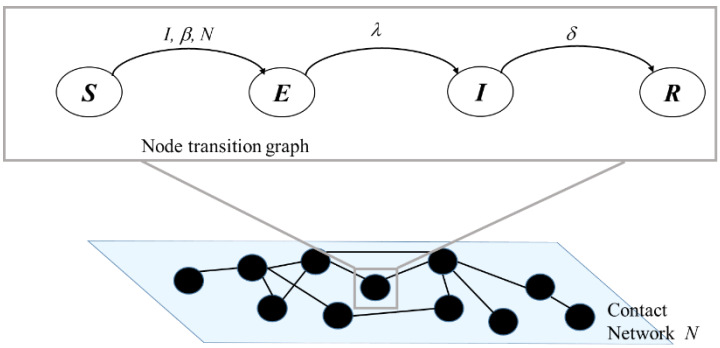

Figure 6. Node transition graph for the SEIR model

SAIS on a single-layer network

The Susceptible-Alert-Infected-Susceptible (SAIS) model was developed to incorporate individual reactions to the spread of the virus. In the SAIS spreading model, each node (individual) can be either *susceptible*, *infected*, or *susceptible-alert*. Hence, the number of compartments in the SAIS model is *M* = 3. The recovery process in SAIS is the same as the recovery process in the SIS model, and is characterized by the recovery rate δ. The infection process of a susceptible agent is also similar to that of the SIS model, which is determined by infection rate β and contact network *N*. However, in the SAIS model, a susceptible node can become alert if it senses infected agents in its neighborhood. In the SAIS model, the alerting transition rate is κ times the number of infected agents. An alert node can also become infected by the process similar to the infection

process of a susceptible node. However, the infection rate for alert nodes is lower due to the adoption of preventive behaviors. The alert infection rate is denoted by β_a *with* $0<\beta_a<\beta$. A table of the main characteristics and a schematic for the SAIS model are shown in the following Table 4 and Figure 7.

Table 4. Descriptors of the SAIS one layer model

Susceptible-Alert-Infected-Susceptible					
States	Transition	Type	Transition parameter	Inducer	Layer
S (susceptible) S_A (susceptible-alert) I (infected)	$(S \rightarrow I)$	edge-based	β	Neighbors in state I	Contact Network N
	$(S \rightarrow S_A)$	edge-based	κ	Neighbors in state I	Contact Network N
	$(I \rightarrow S)$	node-based	δ		
	$(S_A \rightarrow I)$	edge-based	β_a	Neighbors in state I	Contact Network N

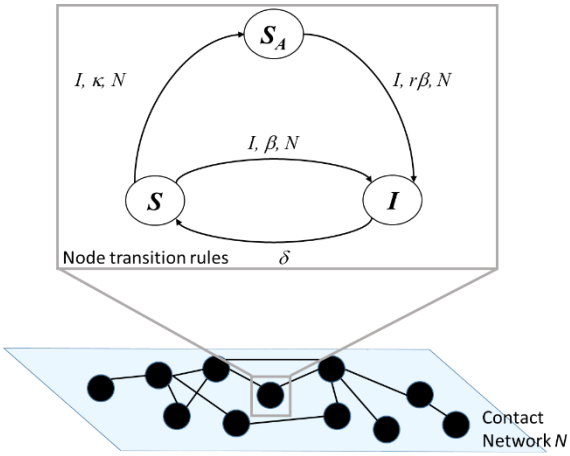

Figure 7. Node transition graph for the SAIS one layer model

SAIS on a two-layer network

The Susceptible-Alert-Infected-Susceptible (SAIS) model on a two layer network was developed to incorporate multiple sources of information to react to the spread of the virus. In the SAIS spreading model, each node (individual) can be either *susceptible*, *infected*, or *susceptible-alert*. Hence, the number of compartments in the SAIS model is $M = 3$. The recovery process in SAIS is the same as the recovery process in the SIS model, and is characterized by the recovery rate δ. The infection process of a susceptible agent is also similar to that of the SIS model, which is determined by infection rate β and contact network N_A. However, in this version of the SAIS model, a susceptible node can become alert if 1) it senses infected agents in its contact neighborhood, or 2) it is notified about the infected neighbors in a notification network N_B. The alerting transition rate is κ times the number of infected agents in the contact network and is μ times the number of infected agents in the notification network. An alert node can also become infected by the process similar to the infection process of a susceptible node. However, the infection rate for alert nodes is lower due to the adoption of preventive behaviors. . The alert infection rate is denoted by β_a *with* $0<\beta_a<\beta$.. A table of the main characteristics and a schematic for the SAIS-2 model are shown in the following Table 5 and Figure 8.

Table 5. Descriptors of the SAIS two layer model

Susceptible-Alert-Infected-Susceptible					
States	Transition	Type	Transition parameter	Inducer	Layer
S (susceptible) S_A (susceptible-alert) I (infected)	$(S \to I)$	edge-based	β	Neighbors in state I	Contact Network N_A
	$(S \to S_A)$	edge-based	κ	Neighbors in state I	Contact Network N_A
	$(S \to S_A)$	edge-based	μ	Neighbors in state I	Contact Network N_B
	$(I \to S)$	node-based	δ		
	$(S_A \to I)$	edge-based	β_a	Neighbors in state I	Contact Network N_A

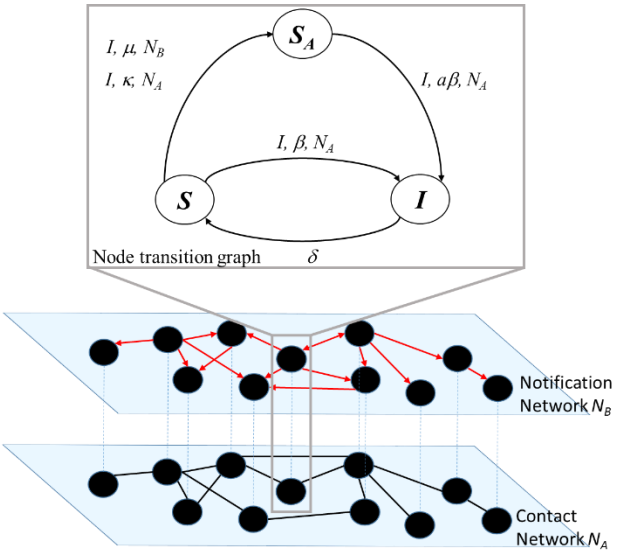

Figure 8. Node transition graph for the SAIS two layer model

SI_1SI_2S on a two-layer network

The SI_1SI_2S model is an extension of continuous-time SIS spreading of a single virus on a simple graph, to the modeling of competitive viruses on a two-layer network. In this model, each node is either *susceptible*, 1-*infected*, or 2-*infected*, i.e., infected by virus 1 or 2, respectively. While virus 1 spreads through network N_1, virus 2 spreads through network N_2. In this competitive scenario the two viruses are exclusive: a node cannot be infected by virus 1 and virus 2 simultaneously. Consistent with SIS propagation on a single layer, the infection and recovery processes for virus 1 and 2 have similar characteristics. The curing process for 1-infected node i is a Poisson process with recovery rate $\delta_1 > 0$. The infection process for susceptible node i effectively occurs at rate $\beta_1 Y_i(t)$, where $Y_i(t)$ is the number of 1-infected neighbors of node i at time t in layer N_1. Recovery and infection processes for virus 2 are similarly described. A table of the main characteristics and a schematic for the SI_1SI_2S model are shown in the following Table 6 and Figure 9.

Table 6. Descriptors of the SI_1SI_2S two layer model

Susceptible-Infected 1-Susceptible- Infected 2-Susceptible					
States	Transition	Type	Transition parameter	Inducer	Layer
S (susceptible) I_1 (infected by virus 1) I_2 (infected by virus 2)	$(S \rightarrow I_1)$	edge-based	β_1	Neighbors in state I	Contact Network N_1
	$(S \rightarrow I_2)$	edge-based	β_2	Neighbors in state I	Contact Network N_2
	$(I_1 \rightarrow S)$	node-based	δ_1		
	$(I_2 \rightarrow S)$	node-based	δ_2		

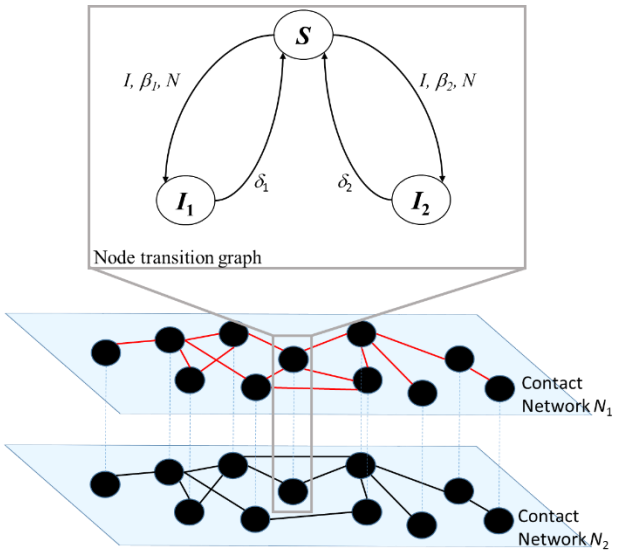

Figure 9. Node transition graph for the SI1SI2S two layer model

Simulator

How is GEMF_Tool organized?

GEMF_Tool is a software tool to create and simulate epidemic models on multilayer networks, written in MATLAB programming language. Several files constitute GEMF_Tool, the most important of which are listed and explained below.

GEMF_main.m

This is the main program of GEMF. First, the problem setup is defined: the multilayer network, the parameter set and the initial conditions. The number of nodes in the network is denoted by N, the network layers are denoted as (Net1, Net2, etc.). The critical information about model characteristics and parameters are in files starting with "`Para`", such as `Para_SIS(delta,beta.`
Second, the stochastic simulation is performed, specifying the length of the simulation and post-processing tasks. `GEMF_SIM(Para,Net,x0,StopCond).` Finally, the set of ordinary differential equations are solved by `GEMF_ODE(Para,Net,X0,T(end))` and final processing of output data is performed.

GEMF_SIM.m

This function simulates one realization of the stochastic Markov process corresponding to the epidemic model. The simulation is event-based, and stops either when the number of events or the simulation time reach a maximum value.

GEMF_ODE.m

This function solves numerically the set of differential equations of GEMF using the the MATLAB library function `ode45(@GEMF_ODE_SOLVER,[0,T],X0_vec)`

Initial_Cond_Gen.m

This function sets the initial conditions of the epidemic, in other words the initial state of each node. This initial state can be assigned as an input, or stochastically defined from a uniform distribution or another given distribution.

NetGen_Import.m

`Net_Import(File,N)` imports a generic network through a text file of the adjacency list of the network, and the total number of nodes in the network.

NetGen_ER.m

`NetGen_ER(N,p)` generates an Erdos-Renyi network with *N* nodes and probability *p* of creating a link among and two nodes.

NetGen_Geo.m

`NetGen_Geo(N,r)` generates a random geometric network with N nodes and connectivity radius r.

Para_Files

The file starting with `Para` defines the number of compartments M=2, assigns an index to each compartment, defines the number of layers N, the inducer compartment in each layer (q=[inducer in layer 1, inducer in layer 2, etc.]), the nodal transition parameters A_d(comp-x,comp-y), the edge-based transition parameter A_b(comp-x,comp-y,layer-z) in each layer.

Using GEMF_Tool

What do I need to start?

To start you only need MATLAB® and our GEMF _Tool. GEMF code is written in MATLAB high level language in such a way that minimizes the use of specific MATLAB functions, allowing also an easy translation of the code in another programming language.

How can I implement an epidemic model?

First step in implementing an epidemic model is to open the file GEMF_main.m and set up the multilayer network. First assign the value to the variable N representing the number of nodes in the network. Second, assign value to the variables Net1, Net2, etc., one for each network layer. Finally define the combination of those layers as your multilayer network `Net=NetCmbn({Net1,Net2, etc.})`.

For example, suppose you want to consider a two-layer network with 1000 nodes, where the first layer is a geometric random network with connectivity radius r, and the second layer is an Erdos-Renyi random network with probability of creating a link p. The following code illustrate this example.

```
% Initial Setup
N=1000;

%Network
r=sqrt(2*log(N)/N);
Net1=NetGen_Geo(N,r);

p=2*log(N)/N;
Net2=NetGen_ER(N,p);

Net=NetCmbn({Net1,Net2});
```

If you want to import your own network with 300 nodes, its adjacency list is in file `mynetwork.txt` and you want to consider only one layer, the example below shows this case.

```
% Initial Setup
 N=300;

%Network

File='C:\Users\GEMFuser\Desktop\GEMF\mynetwork.txt';
N=300;
Net1=Net_Import(File,N);

Net=NetCmbn({Net1});
```

After setting the multilayer network, model parameters need to be defined. Model parameters are closely related to the epidemic model in exam, and can be very different from model to model. In the following, we show how to set parameters and initial conditions for the epidemic model discussed in the precious section of this tutorial.

SIS

Recalling Table 1 that illustrate parameters and inducer states, for the SIS model on a one-layer network, we have:

Susceptible-Infected-Susceptible					
States	Transition	Type	Transition parameter	Inducer	Layer
S (susceptible) I (infected)	$(S \to I)$	edge-based	β	Neighbors in state I	Contact Network N
	$(I \to S)$	node-based	δ		

This table has the information that allow us to define the file Para_SIS.m

```
function Para=Para_SIS(delta,beta)

M=2; q=[2]; L=1;
A_d=zeros(M); A_d(2,1)=delta;
A_b=zeros(M,M,L); A_b(1,2,1)=beta;

Para={M,q,L,A_d,A_b};

end
```

Here, M=2 represents the number of compartments, index 1 representing compartment *susceptible*, index 2 representing compartment *infected*. The vector q=[2] represents the inducer compartments in each layer. In this case we have a single layer, so the vector has a single component and the value is 2 being 2=infected the inducer compartment. The number of layers L in this case is 1, equal to the number of components of vector q.

To set the node-based transition parameter delta, we specify that the component of the matrix A_d equal to delta is the one in position (initial state->final state) of the transition, in this case $(I \to S)$, equivalent to (2,1). This becomes: `A_d(2,1)=delta`. All other components of the matrix are zero: `A_d=zeros(M)`.

To set the edge-based transition parameter beta, we need three indices: one for the initial state, one for the final state, and one for the inducer compartment layer. For this reason, we specify that the component of matrix A_b equal to beta is the one in position initial state=1 (susceptible), final state=2 (infected), and inducer compartment layer 1. This becomes `A_b(1,2,1)=beta`. All other components also of this matrix are equal to zero: `A_b=zeros(M,M,L)`.

After having created the file for the SIS model parameters, we can modify the initial part of GEMF_main.m, adding the numerical values for these parameters. In the specific case illustrated below, we set delta=1 and beta=0.5. We then call the function Para_SIS. Finally, we set as initial condition of the epidemic, randomly selected= Population, infected=2 nodes=10: `x0=Initial_Cond_Gen(N,'Population',[2],[10])`

```
% Parameters and initial conditions

delta=1; beta=0.5;
Para=Para_SIS(delta,beta); M=Para{1};
x0=Initial_Cond_Gen(N,'Population',[2],[10]);
StatesPlot=[1,2];
```

SIR

We define Para_SIR following the SIR Table 2. In this model M=3, with 1 being susceptible, 2 being infected and 3 being recovered. Now delta is the transition rate from the infected state (2) to the recovered state (3). Everything else is the same.

```
function Para=Para_SIR(delta,beta)

M=3; q=[2]; L=length(q);
A_d=zeros(M); A_d(2,3)=delta;
A_b=zeros(M,M,L); A_b(1,2,1)=beta;

Para={M,q,L,A_d,A_b};

end
```

The corresponding new part of GEMF_main.m is as follows, for the specific case of δ=1 and β=0.5:

```
% Parameters and initial conditions

delta=1; beta=0.5;
Para=Para_SIR(delta,beta); M=Para{1};
x0=Initial_Cond_Gen(N,'Population',[2],[10]);
StatesPlot=[1,2];
```

SEIR

We define Para_SEIR following the SEIR Table 3. In this model M=4, with 1 being susceptible, 2 being exposed, 3 being infected and 4 being recovered. Now lambda is the nodal-based transition rate from the exposed state (2) to the infected state (3), and delta is the transition rate from the infected state (3) to the recovered state (4). The inducer compartment here is 3, so `q=[3]`. Everything else is the same.

```
function Para=Para_SEIR(delta,lambda,beta)

M=4; q=[3]; L=length(q);
A_d=zeros(M); A_d(3,4)=delta; A_d(2,3)=lambda;
A_b=zeros(M,M,L); A_b(1,2,1)=beta;

Para={M,q,L,A_d,A_b};

end
```

The corresponding new part of GEMF_main.m is as follows, for the specific case of δ=1, λ=0.3, and β=0.5:

```
% Parameters and initial conditions

delta=1; lambda=0.3; beta=0.5;
Para=Para_SIR(delta,lambda,beta); M=Para{1};
x0=Initial_Cond_Gen(N,'Population',[2],[10]);
StatesPlot=[1,2];
```

SAIS-one layer

We define Para_SAIS_one following the SAIS model on a one-layer network as defined in Table 4. In this model M=3, with 1 being *susceptible*, 2 being *infected*, and 3 being *susceptible-alert*.

The vector q=[2] represents the inducer compartments in each layer. In this case we have a single layer, so the vector has a single component and the value is 2, being 2=infected the inducer compartment. The number of layers L in this case is 1, equal to the number of components of vector q.

To set the node-based transition parameter delta, we specify that the component of the matrix A_d equal to delta is the one in position (initial state->final state) of the transition, in this case ($I \rightarrow S$), equivalent to (2,1). This becomes: `A_d(2,1)=delta`. All other components of the matrix are zero: `A_d=zeros(M)`.

To set the edge-based transition parameter beta, we need three indices: one for the initial state, one for the final state, and one for the inducer compartment layer. For this reason, we specify that the component of matrix A_b equal to beta is the one in position initial state=1 (susceptible), final state=2 (infected), and inducer compartment layer 1. This becomes `A_b(1,2,1)=beta`. The component of matrix A_b equal to kappa is the one in position initial state=1 (susceptible), final state=3 (susceptible-alert), and inducer compartment layer 1. This becomes `A_b(1,3,1)=kappa`. Finally, the component of matrix A_b equal to beta_a is the one in position initial state=3 (susceptible-alert), final state=2 (infected), and inducer compartment layer 1. This becomes `A_b(3,2,1)=beta_a`. All other components also of this matrix are equal to zero: `A_b=zeros(M,M,L)`.

```
function Para=Para_SAIS_one(delta,beta,beta_a,kappa)

M=3; q=[2]; L=length(q);

A_d=zeros(M); A_d(2,1)=delta;
A_b=zeros(M,M,L); A_b(1,2,1)=beta; A_b(1,3,1)=kappa; A_b(3,2,1)=beta_a;

Para={M,q,L,A_d,A_b};
```

After having created the file for the SAIS one-layer model parameters, we can modify the initial part of GEMF_main.m, adding the numerical values for these parameters. In the specific case illustrated below, we set delta=1, beta=0.5, beta_a=0.3, kappa=0.5. We then call the function Para_SAIS_one. Finally, we set as initial condition of the epidemic, randomly selected= Population, infected=2 nodes=10:
`x0=Initial_Cond_Gen(N,'Population',[2],[10])`

```
delta=1; beta=0.5; beta_a=0.3; kappa=0.5;
Para=Para_SAIS_one(delta,beta,beta_a,kappa); M=Para{1};
StatesPlot=[1,2,3];
x0=Initial_Cond_Gen(N,'Population',[2],[10]);
```

SAIS-two layers

We define Para_SAIS_two following the SAIS model on a two-layer network as defined in Table 5. In this model M=3, with 1 being susceptible, 2 being infected, and 3 being susceptible-alert.

The vector q represents the inducer compartments in each layer. In this case we have two layers, so the vector has two components and the value is 2, being 2=infected the inducer compartment in both layers. This becomes `q=[2,2]`. The number of layers L in this case is 2, equal to the number of components of vector q.

To set the node-based transition parameter delta, we specify that the component of the matrix A_d equal to delta is the one in position (initial state->final state) of the transition, in this case (*I*→*S*), equivalent to (2,1). This becomes: `A_d(2,1)=delta`. All other components of the matrix are zero: `A_d=zeros(M);`.

To set the edge-based transition parameter beta, we need three indices: one for the initial state, one for the final state, and one for the inducer compartment layer. For this reason, we specify that the component of matrix A_b equal to beta is the one in position initial state=1 (susceptible), final state=2 (infected), and inducer compartment layer 1. This becomes `A_b(1,2,1)=beta`. The component of matrix A_b equal to kappa is the one in position initial state=1 (susceptible), final state=3 (susceptible-alert), and inducer compartment layer 1. This becomes `A_b(1,3,1)=kappa`. The component of matrix A_b equal to beta_a is the one in position initial state=3 (susceptible-alert), final state=2 (infected), and inducer compartment layer 1. This becomes `A_b(3,2,1)=beta_a`. Finally, the component of matrix A_b equal to mu is the one in position initial state=1 (susceptible), final state=3 (susceptible-alert), and inducer compartment layer 2. This becomes `A_b(1,3,2)=mu`. All other components of this matrix are equal to zero: `A_b=zeros(M,M,L)`.

```
function Para=Para_SAIS_two(delta,beta,beta_a,kappa,mu)

M=3; q=[2,2]; L=length(q);

A_d=zeros(M); A_d(2,1)=delta;
A_b=zeros(M,M,L); A_b(1,2,1)=beta; A_b(1,3,1)=kappa; A_b(3,2,1)=beta_a;
A_b(1,3,2)=mu;

Para={M,q,L,A_d,A_b};
```

After having created the file for the SAIS two-layer model parameters, we can modify the initial part of GEMF_main.m, adding the numerical values for these parameters. In the specific case illustrated below, we set delta=1, beta=0.5, beta_a=0.3, kappa=0.5, and mu=0.5. We then call the function Para_SAIS_two. Finally, we set as initial condition of the epidemic, randomly selected= Population, infected=2 nodes=10: `x0=Initial_Cond_Gen(N,'Population',[2],[10])`.

```
delta=1; beta=0.5; beta_a=0.3; kappa=0.5; mu=0.5;
Para=Para_SAIS_two(delta,beta,beta_a,kappa,mu); M=Para{1};
StatesPlot=[1,2,3];
x0=Initial_Cond_Gen(N,'Population',[2],[10]);
```

SI1SI2S

Recalling Table 6 that illustrates parameters and inducer states for the SI1SI2S model on a two-layer network, we have:

Table 6. Descriptors of the SI_1SI_2S two layer model

Susceptible-Infected 1-Susceptible- Infected 2-Susceptible					
States	Transition	Type	Transition parameter	Inducer	Layer
S (susceptible) I_1 (infected by virus 1) I_2 (infected by virus 2)	$(S \rightarrow I_1)$	edge-based	β_1	Neighbors in state I	Contact Network N_1
	$(S \rightarrow I_2)$	edge-based	β_2	Neighbors in state I	Contact Network N_2
	$(I_1 \rightarrow S)$	node-based	δ_1		
	$(I_2 \rightarrow S)$	node-based	δ_2		

Here, M=3 represents the number of compartments, index 1 representing compartment *susceptible*, index 2 representing compartment *infected by virus 1*, and index 3 representing compartment *infected by virus 2*. The vector q=[2, 3] represents the inducer compartments in each layer. In this case we have two layers, so the vector has two components and one value is 2, being 2=infected-by-virus-1 the inducer compartment in layer 1 and the other value is 3, being 3=infected-by-virus-2 the inducer compartment in layer 2. The number of layers L in this case is 2, equal to the number of components of vector q.

To set the node-based transition parameter delta, we specify that the component of the matrix A_d equal to delta_1 is the one in position (initial state->final state) of the transition, in this case $(I1 \rightarrow S)$, equivalent to (2,1). This becomes: `A_d(2,1)=delta_1.` The component of the matrix A_d equal to delta_2 is the one in position (initial state->final state) of the transition, in this case $(I2 \rightarrow S)$, equivalent to (3,1). This becomes: `A_d(3,1)=delta_2.` All other components of the matrix are zero: `A_d=zeros(M).`

To set the edge-based transition parameter beta, we need three indices: one for the initial state, one for the final state, and one for the inducer compartment layer. For this reason, we specify that the component of matrix A_b equal to beta_1 is the one in position initial state=1 (susceptible), final state=2 (infected by virus 1), and inducer compartment layer 1. This becomes `A_b(1,2,1)=beta_1.` The component of matrix A_b equal to beta_2 is the one in position initial state=1 (susceptible), final state=3 (infected by virus 3), and inducer compartment layer 2. This becomes `A_b(1,3,2)=beta_2` All other components also of this matrix are equal to zero: `A_b=zeros(M,M,L).`

After having created the file for the SIS model parameters, we can modify the initial part of GEMF_main.m, adding the numerical values for these parameters. In the specific case illustrated below, we set delta=1 and beta=0.5. We then call the function Para_SIS. Finally, we set as initial condition of the epidemic, randomly selected= Population, infected=2 nodes=10: `x0=Initial_Cond_Gen(N,'Population',[2],[10])`

```
function Para=Para_SI1SI2S(delta_1,delta_2,beta_1,beta_2)

M=3; q=[2,3]; L=length(q);
A_d=zeros(M); A_d(2,1)=delta_1; A_d(3,1)=delta_2;
A_b=zeros(M,M,L); A_b(1,2,1)=beta_1; A_b(1,3,2)=beta_2

Para={M,q,L,A_d,A_b};

end
```

After having created the file for the SI1SI2S two-layer model parameters, we can modify the initial part of GEMF_main.m, adding the numerical values for these parameters. In the specific case illustrated below, we set delta_1=1, delta_2=1, beta_1=0.5, beta_2=0.3. We then call the function Para_SI1SI2S. Finally, we set as initial condition of the epidemic, randomly selected= Population, infected=2 nodes=10:
`x0=Initial_Cond_Gen(N,'Population',[2],[10]).`

```
delta_1=1; delta_2=1;beta_1=0.5; beta_2=0.3;
Para=Para_SI1SI2S(delta_1,delta_2,beta_1,beta_2); M=Para{1};
StatesPlot=[1,2,3];
x0=Initial_Cond_Gen(N,'Population',[2],[10]);
```

How can I implement my new epidemic model?

Any other model can be easily implemented following the procedure we have explained to create a variety of models above.

We conclude this tutorial summarizing the steps needed to adapt GEMF to your own epidemic model.

Step 1 – Define the network and its layers

Step 2 – Define the node transition graph

Step 3 – Define the model's parameter table

Step 4 – Create the file Para

Step 5 – Modify GEMF_main.m to input the network and the parameters correctly.

Concluding remarks

This work has been supported by the National Science Foundation under award CIF-1423411. Any opinions, findings, and conclusions or recommendations expressed in this website are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] F. Darabi Sahneh, C. Scoglio, P. Van Mieghem, "Generalized Epidemic Mean-Field Model for Spreading Processes over Multi-Layer Complex networks," IEEE/ACM Transactions on Networking, vol. 21, no.5, 1609-1620, 2013.