

Homework Program 6: An Enhanced SIR Markov Chain Model

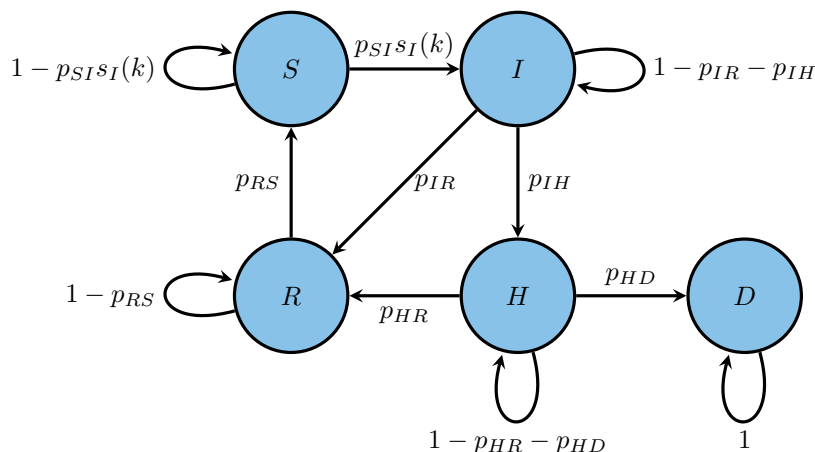


Figure 1: SIR model with additional states “H” (Hospitalized) and “D” (Deceased).

In this assignment you will study an extended version of the SIR model discussed in the second case study. Specifically, in addition to the states “Susceptible”, “Infected”, and “Recovered”, we will add the states “Hospitalized” and “Deceased” to represent the fraction of the population that are hospitalized at a given time, and the cumulative fraction that, sadly, die. The assignment will consist of coding the model and using it to evaluate the effects of deterrence methods that can potentially slow the spread of an infection and reduce hospitalizations and deaths.

The Markov model is shown in Figure 1 where the nodes represent the states (“S” for susceptible, “I” for infected, and so forth), and the state transitions are labeled with the associated probability. For example, p_{HR} is the probability, or fraction of the population currently in state H (hospitalized) that recover, and so transition to state R.

Task 1: Write the set of update equations for the state variables in matrix-vector form: $\mathbf{s}(k+1) = \mathbf{P}'\mathbf{s}(k)$.

For example, from the diagram we have for state I:

$$s_I(k+1) = p_{SI}s_I(k)s_S(k) + (1 - p_{IR} - p_{IH})s_I(k) \quad (1)$$

where $s_I(k)$ can be interpreted as the fraction of the population that is infected at time step k . Note that the coefficient multiplying $s_S(k)$ is $p_{SI}s_I(k)$, which depends on $s_I(k)$. This means the model is nonlinear (the coefficient changes with the state); however, we can still write the update for the state as $\mathbf{s}(k+1) = \mathbf{P}'\mathbf{s}(k)$, where the corresponding element of \mathbf{P}' is $p_{SI}s_I(k)$.

Write the analogous updates for the other state variables and define the probability transition matrix \mathbf{P} , which you will need to perform the subsequent tasks. Define the state vector as: $\mathbf{s} = [s_S \ s_I \ s_H \ s_R \ s_D]'$, that is, use this specific ordering for the state variables to construct \mathbf{P} . *You do not have to submit your derivation for this task, but be sure to indicate clearly with a comment in your code for Task 2 where the matrix \mathbf{P} is defined.*

Task 2: Write an initial version of the function `sihrd`, which outputs the sequence of state values $\mathbf{s}(k)$ at times $1, \dots, K$ as a $5 \times K$ array and plots each state variable as a function of k . The inputs are:

- `steps` is the total number of time steps K . That is, the function will compute $\mathbf{s}(k)$ for $k = 1, \dots, K$.
- `s_init` is the initial state vector at $k = 1$ ordered as in Task 1.
- `param` is an array of size `[1 6]` containing the six model parameters: p_{SI} , p_{IH} , p_{IR} , p_{HR} , p_{RS} , and p_{HD} , in this order.

The function should check that:

- all three inputs are present,
- `steps` is a positive integer,
- the initial state vector `s_init` has the correct size with nonnegative elements,
- the parameter array `param` has the correct size with elements between (or equal to) 0 and 1.

To use the `arguments` block to check the size and elements of `param`, you can include the statement `param(1,:) {mustBeInRange(param,0,1),mustHaveLength(param,6)}` where `mustHaveLength` is a function that can be included at the end of the `sihrd` function:

```
function mustHaveLength(x,L)
    if length(x) ~= L
        eid = 'Size:badLength';
        msg = 'The input length is %u but it should be %u.';
        error(eid,msg,length(x),L);
    end
end
```

The function `sihrd` should then print the initial state values and parameters as shown in the example output below. An easy way to do this is with `fprintf` using `%s` to print an array of strings, i.e., `["S","I","H","R","D"]` for the states. Note the use of double-quotes.

The function should generate two figures, one showing the trajectories of the state variables s_I , s_H , and s_D , and the other showing s_S and s_R . (This looks better and is easier to examine than showing all five plots on one figure. Note that you can use `subplot` to view the figures side-by-side.) Run an example with `steps = 600`, the following parameters, and answer the questions.

The parameters are:

```
p_SI: 0.0500
p_IH: 0.0100
p_IR: 0.0200
p_HR: 0.1000
p_RS: 0.0000
p_HD: 0.0030
```

The initial state probabilities are:

```
S: 0.9500
I: 0.0500
H: 0.0000
R: 0.0000
D: 0.0000
```

- What is the maximum percentage of the population that is hospitalized? Give a brief qualitative explanation (no more than 3 sentences) why the infection and hospitalized states have the hump-shaped trajectories shown.
- Why is s_D always increasing? Explain in terms of the properties of the Markov chain (one sentence).
- Change p_{SI} to 0.1 and rerun the example. Why does the percentage of the population that remain susceptible (that is, never get the disease) decrease?

Task 3: In this Task you will modify the function in Task 2 to take into account deterrence measures that are put into effect once the infection rate starts to rise. We will model this by changing the value of `p_SI` over time, so that in the matrix \mathbf{P} , the constant parameter p_{SI} is replaced by $p_{SI}(k)$, and will depend on what deterrence measures are followed at time k . When a Markov chain parameter varies with the time period, we say that the Markov chain is *time-varying*.

Add the following input to the function `sihrd`:

`deter` is an array of size `[1 steps]` that will be used to change the parameter p_{SI} over time. Specifically, the scalar parameter p_{SI} at time k will be replaced by `p_SI = param(1)*deter(k)`, where `param(1)` corresponds to the value of p_{SI} with no deterrence measures, and all elements of `deter` are between 0 and 1, where 1 signifies no deterrence measures, and 0 signifies that spreading is completely suppressed for that period. In this way we can account for the effects of different levels of deterrence measures during different time periods.

Check that the array `deter` entered by the user has size `(1, steps)`, and that the values are between (or equal to) 0 and 1. If you use the `arguments` block, you can include the analogous statement as for `param`. If `deter` is not specified, then assign a default value of all ones, which corresponds to no deterrence measures. Use the modified function with the parameters in the preceding task to generate plots associated with the following scenarios and answer the questions.

Suppose deterrence measures are followed soon after the infection rate starts to rise, continue for some period, but then are loosened before the infection rate has sufficiently subsided, potentially giving rise to a second wave. As a particular example, plot the trajectories of the state variables where the first 30 elements of `deter` are 1 (no deterrence), the next M (variable) elements are 0.2 (strong deterrence), the next 100 elements are 0.4 (looser deterrence), and the remaining elements are 1 (no deterrence). Take $M = 100$ and explain qualitatively the shape of the infection and hospitalization curves in terms of the input parameters. (You may need to increase the number of steps.) Experimenting with different values of M , what do you conclude about the effect of deterrence measures in preventing a second wave of infections? Does the model show that those measures always help? What may help to mitigate the second wave shown that is not captured in the model?

Note: this example is not meant to correspond to any real-world scenario! In general, using a model such as this one to make forecasts requires that all parameters be *calibrated* using available data or side-information to estimate what they actually are for the particular scenario being studied. Here the intent is to see how the model behaves given certain choices of inputs.

Submission instructions: Submit this assignment as a single file consisting of: (1) The function `sihrd` including the modifications (additional input and argument check) in Task 3, and (2) answers to all questions in the order they appear. You do not need to submit your plots.