

PlantSIR

February 28, 2025

1 Overview

The model described in the provided code simulates the spread of an invasive plant pathogen or pest across a fixed grid, and it can be optimised to match a reference data using a differentiable agent-based model (ABM). The infection dynamics are governed by various parameters that affect transmission rates, recovery rates, and the influence of environmental factors. The model is implemented using PyTorch for efficient computation, and the parameters are optimised through a gradient-based approach.

Our approach is particularly valuable for understanding the spread of pests or pathogens on fixed landscapes, such as in agricultural or woodland-management settings where crops or plants are vulnerable to disease outbreaks or the spread of invasive pests. In this context, the landscape can be treated as a grid where each cell represents a spatial unit that may be susceptible to infection, while environmental factors like plant/crop density, temperature, prevailing winds etc. can influence the dynamics of the spread. The ability to simulate and optimise these dynamics enables the prediction of disease/pest progression, evaluation of control strategies, and development of targeted interventions in crop protection or pest management.

The choice of the SIR (Susceptible-Infected-Recovered) model in this context is motivated by its widespread use in epidemiological studies, particularly in modelling the spread of infectious diseases among populations. While the SIR model is typically applied to human or animal diseases, it has been suggested for use in the spread of invasive species and we have previously adapted it for the spread of the invasive pest Oak Processionally Moth in the UK. The simplicity and versatility of the SIR model make it an appealing choice for such studies, even though it was not strictly derived for plant pathogens or pests. Invasive species, whether pests or diseases, often exhibit a rapid rate of spread and can have significant ecological or economic impacts. The SIR model captures key features of these processes, including the transition from a susceptible state to an infected state, and the eventual recovery or resolution of the infection/infestation. From hereon we will use the terms ‘infected’ and ‘recovered’ as these are standard in the epidemiological literature and suitable for describing the spatial spread of a plant/crop pathogen. If the model is used for the spread of an invasive pest one could think of I as **infested** and the state R as **removed** or **resistant** depending on the problem.

The SIR framework is particularly appealing when compared to a ‘particle’-based approach. Indeed, there is a significant computational advantage of using the model described due to the scalability and numerical stability in the case of large populations or extensive spatial domains, such as those encountered in the simulation of invasive pests. In a particle-based approach, where individual agents (representing pests) are

explicitly tracked, the number of agents in the system typically grows exponentially over time, especially if the agents reproduce or spread across the landscape. This leads to significant numerical challenges, including memory and computational overheads.

The adoption of an agent-based model in the provided code allows for more granular control over spatial dynamics and interactions between individual units, further enhancing the model’s ability to simulate realistic spread patterns. Infection dynamics are governed by parameters that affect transmission rates, recovery rates, and the influence of environmental factors. The model is implemented using PyTorch for efficient computation, and the parameters are optimised through a gradient-based approach. Through optimisation, the model can be adjusted to match observed infection patterns, helping researchers and practitioners make data-driven decisions to mitigate pest or pathogen impacts.

2 Model Components

2.1 Grid Initialisation

The grid represents a two-dimensional space with each cell representing an individual. Each individual can be in one of three compartments:

- **Susceptible (S)**: Individuals not yet infected.
- **Infected (I)**: Individuals currently infected.
- **Recovered (R)**: Individuals who have recovered from the infection.

At the start of the simulation, an initial infection map I_0 is provided, which defines the initial distribution of infected individuals ($I = I_0$). Typically this would be from observational data at the start of the infection. The grid is represented as a tensor **grid** with three channels, one for each compartment. The susceptible individuals are initialised to $S = 1 - I_0$, and the recovered individuals are initialised to $R = 0$.

2.2 Force of Infection Calculation

The force of infection ζ for each cell is computed based on the infection status of its neighbours and the infection transmission weights. The transmission weights are computed using a function that depends on the distance to the neighbours and environmental factors such as plant density. We define a (sparse weight) matrix for the i^{th} cell W_i through:

$$W_{ij} = W(r_{ij}) = \beta \exp \left\{ - \left(\frac{r_{ij}}{\sigma} \right)^\alpha \right\} \quad (1)$$

where r_{ij} is the distance between cell i and its neighbour j and β and σ are parameters controlling the strength and functional form of the transmission. For example $\alpha = 1$ defines an exponential interaction kernel and $\alpha = 2$ a Gaussian kernel.

We can all an anisotropic models where the infection spread is influenced by direction, an additional factor is introduced to weight the effective distance such that:

$$W_{ij} = W(r_{ij}^*, \alpha, \beta, \sigma), \quad r_{ij}^* = r_{ij} / (1 + V |\cos \theta|) \quad (2)$$

where V acts as a imposed ‘velocity’, and θ is the angle between a reference direction ϕ and the direction of infection spread and the vector from cell i to neighbour j . Typically

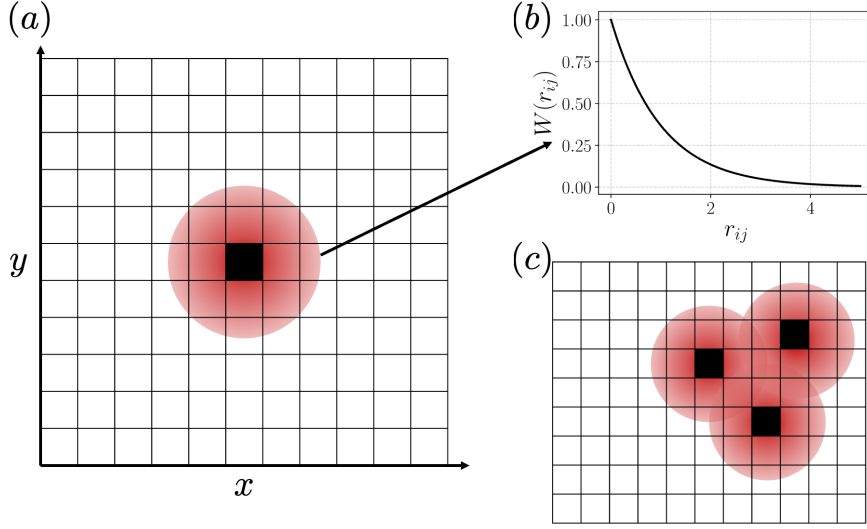


Figure 1: A schematic of spatial spread of ‘infection’ in the model. Panel (a) shows the grid, with one infected cell (I) coloured black at the centre of the grid surrounded by susceptible cells (S - white). Red sketches the spatial interaction between an infected and susceptible cell, which is illustrated in functional form in panel (b). Panel (c) shows how the sum of overlapping interaction kernels arising from multiple infected cells is combined through E.q. 3

ϕ would be a reference direction (for example the prevailing wind direction at the dispersal stage of the pest/pathogen) however this can also be ‘learned’ through parameter optimisation (see below). The effect of this is to promote the spread in directions parallel (and anti-parallel) to ϕ as can be seen in the schematic in Fig. 2.

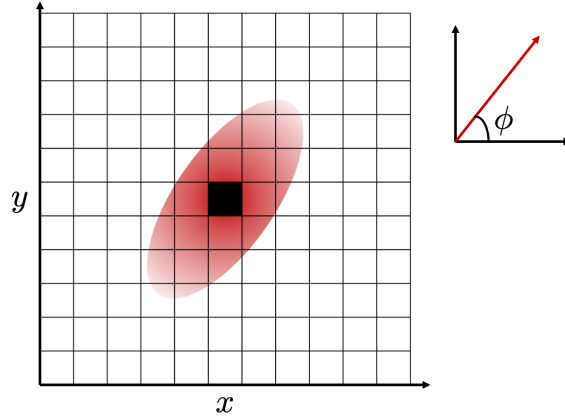


Figure 2: A schematic of anisotropic spread as defined formally by E.q. 2. Here ϕ is a reference direction (prevailing wind etc.) and spread is promoted in this direction by the size of V .

The infection force ζ_i at each cell i is calculated as the weighted sum of the infection status of over all cells in the domain, excluding the self interaction:

$$\zeta_i = \sum_{j, i \neq j} W_{ij} I_j \quad (3)$$

where I_j is the infection status of neighbour j .

This force of infection ζ can also be modified by environmental data. At present we account for the local coverage of the host species at each grid point mapping the environmental data to a value within the interval $[0, 1]$ using a *sigmoid function*. We represent the coverage of the susceptible host (e.g. crop, plant, tree) at the i^{th} grid cell as \mathcal{T} we then incorporate this by modifying the expression for ζ_i :

$$\zeta_i = \omega(\mathcal{T}) \sum_{j, i \neq j} W_{ij} I_j$$

where

$$\omega(\mathcal{T}) = \frac{1}{1 + \exp\{-\rho_0(\mathcal{T} - \rho)\}}$$

Here, ρ_0 is a scaling parameter that controls the steepness of the sigmoid function, and ρ is a threshold value. This effectively limits the spread of the infestation/infection to grid cells with $\mathcal{T}_i > \rho$ and we can interpret ρ as a critical ‘habitability’ value; ρ_0 controls the sharpness of the transition with larger values meaning a rapid transition from a suitable to unsuitable susceptible cell.

2.3 Infection and Recovery Dynamics

The infection dynamics are modelled using the following processes:

- **Infection (S \rightarrow I):** Susceptible individuals can become infected with probability P_{inf} :

$$P_{\text{inf}} = 1 - \exp(-\zeta)$$

where ζ is the force of infection.

- **Recovery (I \rightarrow R):** Infected individuals can recover with probability P_{rec} :

$$P_{\text{rec}} = \gamma$$

where γ is the recovery rate.

2.4 Gumbel-Softmax Approximation for State Transitions

To model the transitions in a differentiable way, we use the Gumbel-Softmax trick. Given a probability P for a transition (e.g., P_{inf} for infection or P_{rec} for recovery), the standard approach would be to sample from a Bernoulli distribution:

$$Z \sim \text{Bernoulli}(P).$$

However, this sampling is not differentiable, which poses issues for gradient-based optimisation. Instead, we employ the Gumbel-Softmax approximation, which enables reparameterisable sampling. The Gumbel distribution is used to transform uniform random variables into samples that approximate categorical sampling. If $U \sim \text{Uniform}(0, 1)$, then the Gumbel-distributed random variable is defined as:

$$G = -\log(-\log U).$$

This allows us to introduce randomness in a way that can be back-propagated.

Gumbel-Softmax Approximation To approximate a Bernoulli sample, we define two logits: one for the event happening (P) and one for the event not happening ($1 - P$). We introduce independent Gumbel noise G_1 and G_2 and compute:

$$\tilde{P} = \frac{\exp((\log P + G_1)/\tau)}{\exp((\log P + G_1)/\tau) + \exp((\log(1 - P) + G_2)/\tau)}$$

where τ is a temperature parameter that controls the degree of approximation. As $\tau \rightarrow 0$, \tilde{P} approaches a hard threshold at 0 or 1.

For state transitions, such as $S \rightarrow I$ (infection), we replace the discrete Bernoulli sample with:

$$\xi = \tilde{P}_{\text{inf}} S.$$

Similarly, for recovery:

$$\eta = \tilde{P}_{\text{rec}} I.$$

These terms are used in the continuous update equations:

$$\Delta S = -\xi S, \quad \Delta I = \xi S - \eta I, \quad \Delta R = \eta I.$$

This formulation ensures differentiability, enabling back-propagation and optimisation via gradient descent.

2.5 Optimisation Process

The optimisation process aims to find the best parameters that minimise the discrepancy between the simulated infection map and a reference infection map. The parameters include the parameters in the spatial transmission kernel (α, β, σ), recovery rate (γ), and advective ‘velocity’ (V) and direction ϕ . The optimisation is performed using the Adam optimiser and a loss function, which can be chosen from various options in the code. For example we can compare the fit between a simulated and reference infection map through the structural similarity index measure (SSIM), a simple mean squared error (MSE).

The loss function is computed as:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (I_{i,\text{sim}} - I_{i,\text{ref}})^2$$

or the Dice loss function (and variants of this):

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2 \sum_i (I_{i,\text{sim}} \cdot I_{i,\text{ref}})}{\sum_i I_{i,\text{sim}} + \sum_i I_{i,\text{ref}}}$$

where I_{sim} and I_{ref} are the simulated and reference infection maps. The Adam algorithm then updates the parameters as follows:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\Theta} \mathcal{L} \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\Theta} \mathcal{L})^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\ \Theta_t &= \Theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \end{aligned}$$

where m_t and v_t are biased first and second moment estimates, η is the learning rate, and ϵ is a small constant for numerical stability.

2.6 Numerical Simulations and Results

The model is run for a set number of timesteps $n_{\text{timesteps}}$. For each timestep, the force of infection and the transition probabilities are updated, and the grid is evolved according to the infection and recovery dynamics. The model is then evaluated against the reference infection map using the loss function.