

Jewell Implementation Version 1

Erica Billig

December 8, 2014

1 Data Simulation

To start, a simple susceptible-infected-notified-removed model was created. It was assumed that the rate of infection was the same among infected and notified houses (ie. there is no decrease in transmission once a house is notified).

A dataset of 500 houses was simulated uniformly on a $[0,1] \times [0,1]$ grid. The entire outbreak was recorded, including the initial infective house, all infection times, notification times, and recovery times. All infection times occur before notification times by simulation set-up. The euclidean distance between each house i and j is recorded as $distance_{i,j}$. The transmission function was set up as follows:

$$\beta_{i,j} = \begin{cases} \beta & \text{if } distance_{i,j} < 0.3 \\ \beta * .1 & \text{if } distance_{i,j} \geq 0.3 \end{cases}$$

An initial infective was randomly chosen. Define Not_i , I_i , and R_i as the notification, infected, and recovery times of house i . Define $D_i = Not_i - I_i$.

$$D \sim \exp(r)$$

The time from notification to recovery $R_i - Not_i \sim \exp(1)$. These distributions could be easily extended to another distribution, such as a gamma distribution.

A sample path is shown in Figure 1 at four time points.

For the first try, I assumed the entire epidemic was followed (until the number of houses in the infected and notified states were both zero). Thus in this case, $T_{obs} = \max(t)$. The unknown parameters for estimation are β , r , and the infection times, I_i .

The likelihood of this problem is shown below:

$$L(I, N, R \mid r, \beta) \propto \prod_{j \neq k}^{N_I} \left(\sum_{i \in Y_{j-}} \beta_{i,j}(I_j) \right) \times \exp\left(- \int_1^{T_{obs}} \sum_{i \in I} \sum_{i \in S} \beta_{i,j}(t - I_i) dt\right) \times \prod_{i=1}^{N_I} f_D(Not_i - I_i)$$

where $Y_{j-} = \{i : I_i < I_j \leq R_i\}$

2 MCMC Algorithm

A brief overview of the MCMC algorithm is as follows:

1. Chose i uniformly at random and update $I_i \mid I_{i-}, r, \beta, Not, R$.

2. Update $r \mid I, \beta, Not, R$
3. Update Z and hence A and B using non-centering*
4. Update $r \mid \beta, U^A, I^B, R$ using the partially non-centered algorithm. Note: this also updates I^A .
5. Update $\beta \mid r, I, Not, R$.

2.1 Non-centering algorithm

Each iteration, a portion of the observations are treated as non-centered in (3) and (4). When non-centered, the variable is transformed so that

$$I_i = Not_i - 1/r * U_i$$

where $U_i \sim \exp(1)$. The random variable Z decides which observations are treated as centered ($Z = 0$) vs. non-centered ($Z = 1$). The paper suggested that non-centering 25% of observations yielded good results. Non-centered observations are put in group B, and centered observations are put in group A.

3 Results

True values set to $\beta_0 = 0.5$ and $r = 1.0$

$$\hat{\beta} = 0.53 \quad \hat{r} = 1.35$$

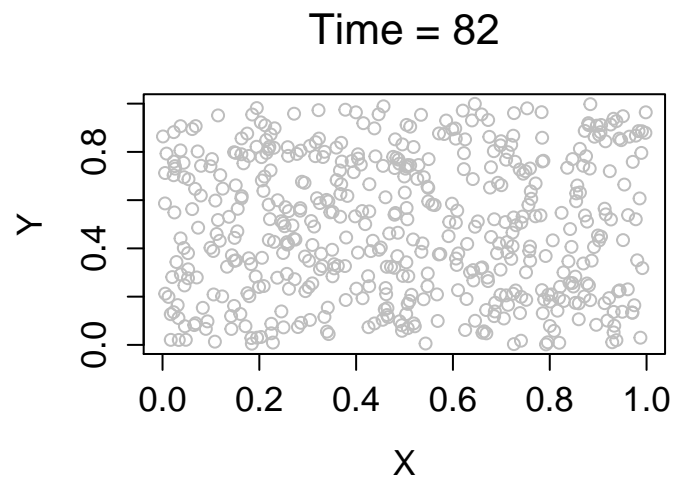
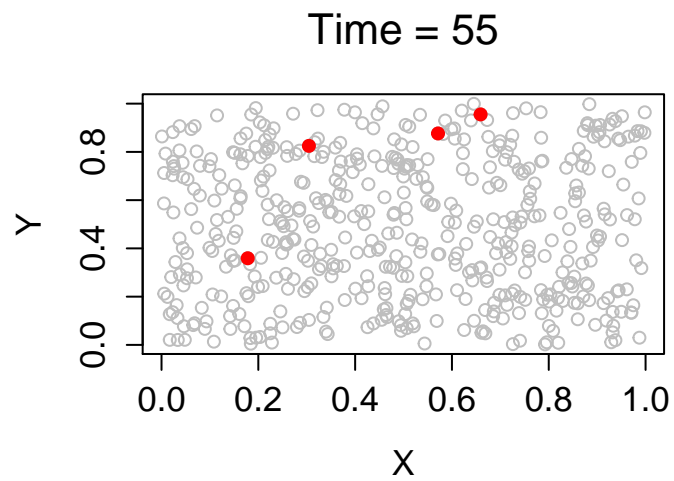
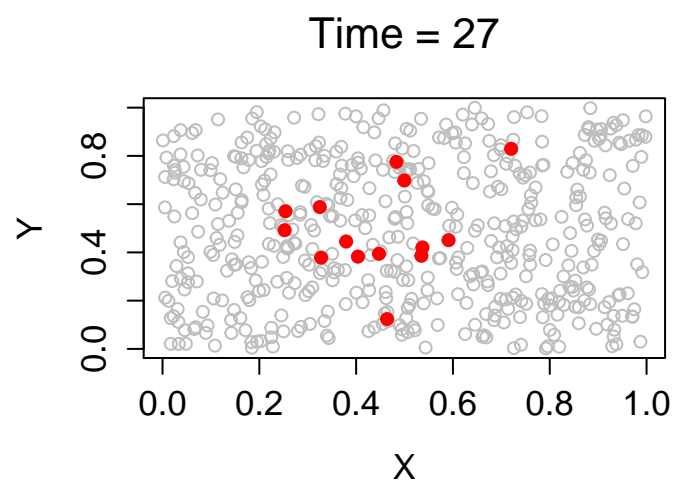
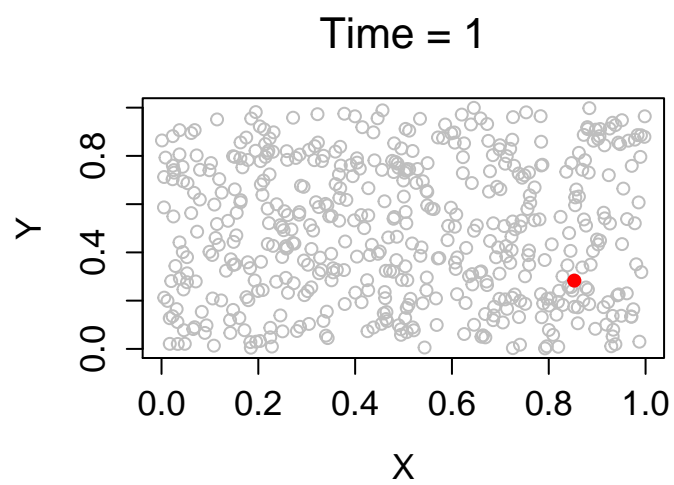


Figure 1: Sample Path

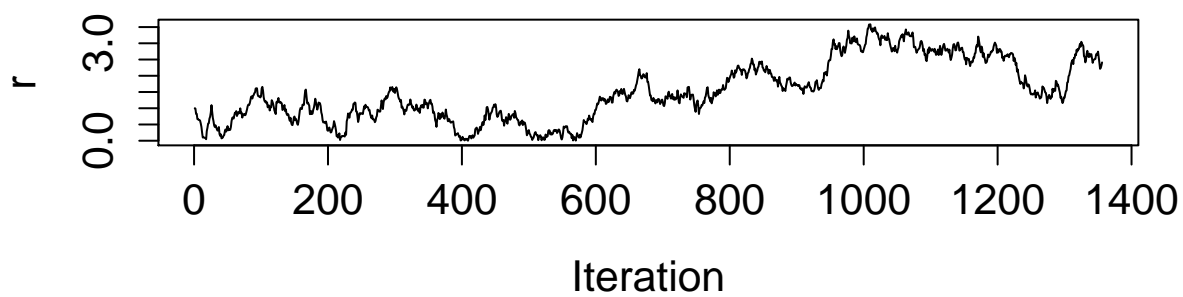
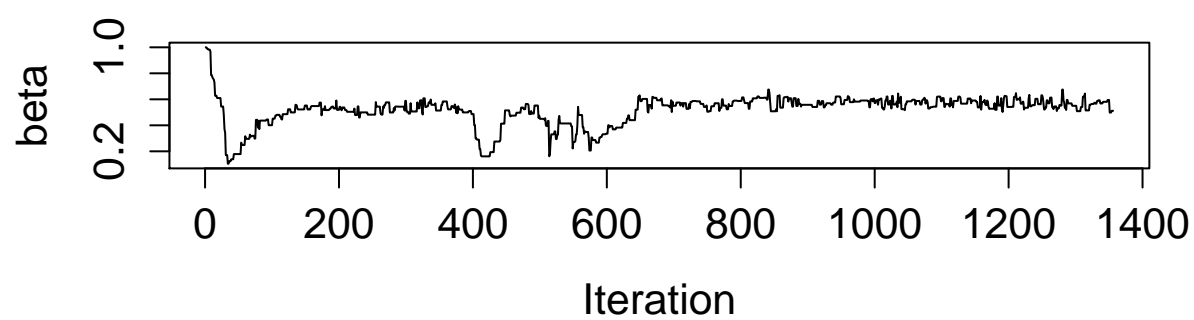


Figure 2: trace plot