

# TMA4250 Spatial Statistics Exercise 1, Spring 2019

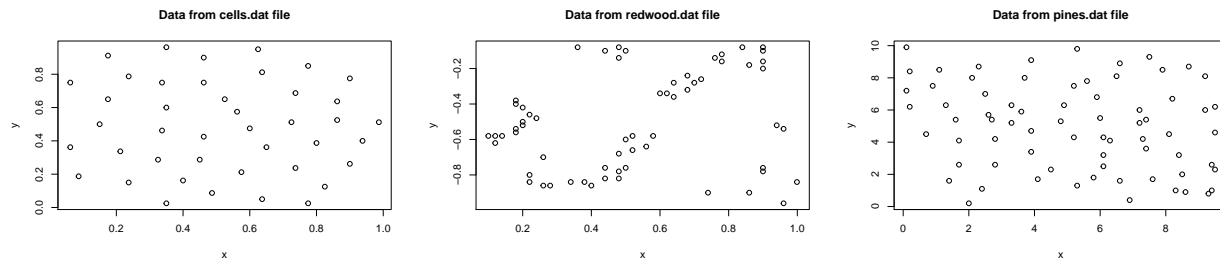
Group members: Henrik Syversveen Lie, Øyvind Auestad

24.03.2019

## Problem 1: Analysis of Point Patterns

a)

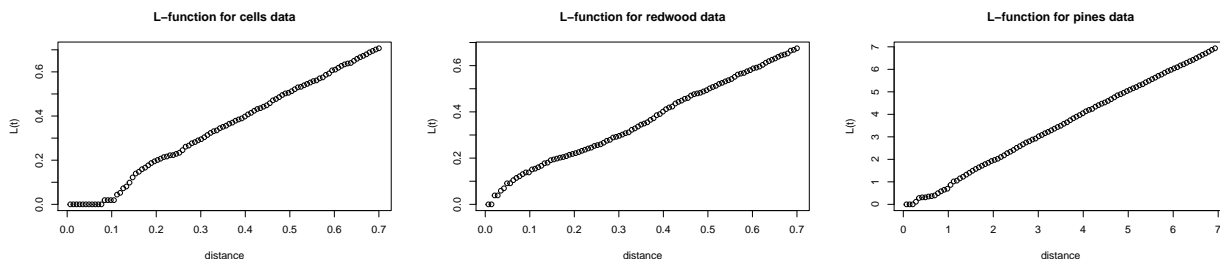
We consider the three real data point patterns found in the R package MASS.



The cell data in the left plot represents some cell structure. Here, the cells have a repulsive effect on each other, so two cells are not likely to be close to each other. Thus, we get a point pattern that looks to be repulsive. The middle plot represents locations of redwood trees. The trees release seeds, which fall to the ground close to them, so trees get clustered together around a “mother tree”. We therefore get a point pattern that contains clusters of points. Finally, the plot to the right is locations of pine trees. Presumably, seeds from these trees can get carried further away from the tree trunk by e.g. animals eating the pinecones. Hence, the location of one tree is independent of locations of other trees, and we get a pattern that looks like a stationary Poisson random field.

b)

We then want to compute the empirical  $L$ -function for each of the point patterns, and use the R function `Kfn()` to do this. The  $L$ -functions are then displayed for each point pattern.



The plots of the  $L$ -functions are closely related to the displayed point patterns. The cells data has a low value of the  $L$ -function for low values of  $t$ . This means that there is a low probability of points being close to each other, we have a repulsive model, which is verified on the displayed point pattern. The redwood data on the other hand, has a high value of the  $L$ -function for low values of  $t$ . This means that there is a large probability of points being close, we have a clustered model, which also is verified on the displayed point pattern. Finally we have the pines data, for which the  $L$ -function closely resembles a straight line. This means that the point pattern is similar to a stationary Poisson RF.

The interaction function  $J(t)$  is defined as,

$$J(t) = \frac{\mathbf{E}[k_{\mathbf{B}_{\mathbf{x}_0}(t)} - 1]}{|\mathbf{B}_{\mathbf{x}_0} \cap D|}; t \in \mathbb{R}_{\oplus},$$

where  $\mathbf{B}_{\mathbf{x}_0}(t)$  is defined as the ball with radius  $t$  and center  $\mathbf{x}_0 \in \mathbb{X}_D$ . For a stationary Poisson random field, one gets,

$$J(t) = \frac{\lambda_k |\mathbf{B}_{\mathbf{x}_0} \cap D|}{|\mathbf{B}_{\mathbf{x}_0} \cap D|} = \lambda_k.$$

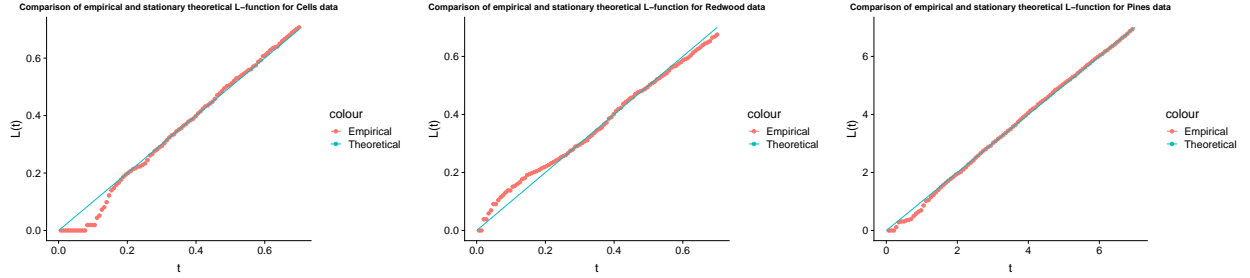
Hence, for stationary Poisson RF, the interaction-function is independent of the distance  $t$ .

An alternative interaction function is the dimension dependent  $L$ -function, which in  $\mathbb{R}^2$  is,

$$L_2(t) = \left[ \frac{\mathbf{E}[k_{\mathbf{B}_{\mathbf{x}_0}(t)} - 1]}{\lambda_k \pi} \right]^{1/2} = J(t) \times \frac{t}{\lambda_k}.$$

For the stationary Poisson random field, we get  $L_2(t) = t$ .

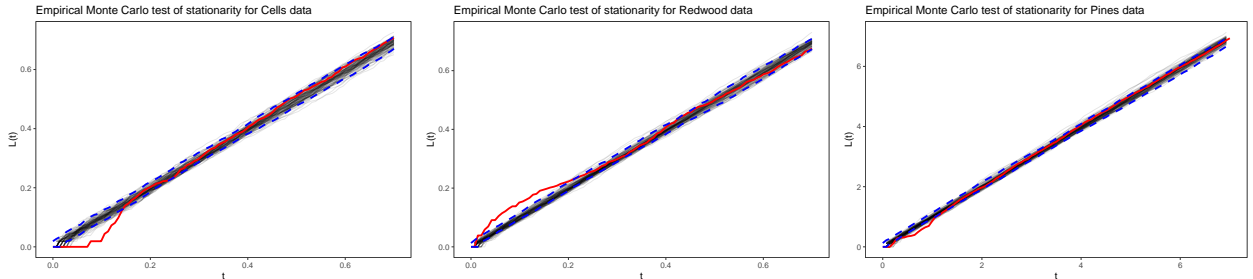
We then display the empirical  $L$ -function for each point pattern with the theoretical function for a stationary Poisson RF.



From the plots, we see that the empirical  $L$ -function for the Pines data is closest to the theoretical  $L$ -function of a stationary Poisson RF. The empirical  $L$ -function of the Redwood and Cells data have large deviations for small values of  $t$ , with Cells data having a smaller value, and Redwood data having a larger value than the theoretical stationary  $L$ -function. From the  $L$ -function and point pattern plots, we hypothesize that a stationary Poisson RF appear as a suitable model for the Pines data, but not for the two others.

c)

Finally, we perform an empirical Monte Carlo test to check whether the stationary Poisson RF is a suitable model. We consider each point pattern - under the hypothesis that the point pattern originates from a stationary Poisson random field, condition on the observed point count, and generate 100 realizations of a stationary Poisson RF. For each realization we compute the  $L$ -function, and use the 100 generated  $L$ -functions to empirically test whether the observed point pattern could originate from a stationary Poisson RF. To do this, we display empirical 0.9-intervals for the  $L$ -functions together with the estimated function for each point pattern.



From the plots, we see that none of the point patterns are inside the 0.9-intervals for all values of  $t$ . Still, the

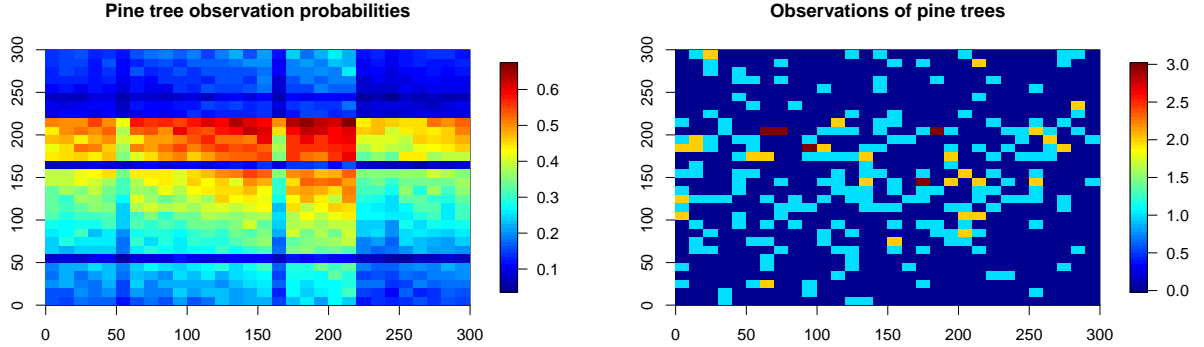
Pines data is the one that is closest to the 0.9-interval. From both the plots of the point patterns and this test, we conclude that the first two (Cells and Redwood) are definitely not from a stationary Poisson RF, while the last (Pines) point pattern could come from a stationary Poisson RF, even though the  $L$ -function is outside the 0.9-interval for some  $t$ .

## Problem 2: Bayesian inversion in Poisson RF

We consider an area of size  $(300 \times 300)$  m<sup>2</sup> containing a pine tree forest, and the actual locations of the pine trees are to be assessed. The pine tree locations are observed from a satellite by remote sensing, and due to partly cloudy weather the observation probability for individual trees vary across the area. We discretize the area into a regular  $(30 \times 30)$ -grid  $L$  with grid unit size  $\Delta_n = 100\text{m}^2$ . The true, but unknown, number of pine trees located in each grid unit is the  $n$ -vector  $\mathbf{k}$ , with  $\{k(\mathbf{x}); \mathbf{x} \in L\}$ . The probabilities for observing a tree in each grid unit is represented by the  $n$ -vector  $\{\alpha(\mathbf{x}); \mathbf{x} \in L\}$ , which vary across the area. We also denote by the  $n$ -vector  $\{d(\mathbf{x}); \mathbf{x} \in L\}$  the number of trees observed in each grid unit.

a)

First, we display the observations and the observation probabilities across the grid. The observation probabilities vary from 0.04 to 0.67, and the number of observed trees in each grid unit varies between 0 and 3.



We further assume that the observations given the true number of trees are spatially uncorrelated from one grid unit to another. We then specify the likelihood model for the observations. For each grid unit  $i$  one has a binomial likelihood model with  $d_i \leq k_i$ ,

$$[d_i | k_i] \sim p(d_i | k_i) = \binom{k_i}{d_i} \alpha_i^{d_i} [1 - \alpha_i]^{k_i - d_i}.$$

Because the observations are conditionally independent from one node to another, the likelihood model with  $\mathbf{d} \leq \mathbf{k}$  is,

$$[\mathbf{d} | \mathbf{k}] \sim p(\mathbf{d} | \mathbf{k}) = \prod_{i=1}^n p(d_i | k_i) = \prod_{i=1}^n \binom{k_i}{d_i} \alpha_i^{d_i} [1 - \alpha_i]^{k_i - d_i}.$$

b)

To further specify our model, we assume apriori that the distribution of trees is according to a stationary Poisson RF with model parameter  $\lambda_k$ . The prior Poisson model is defined on the discretized event count

representation  $\mathbf{k} \in \mathbb{N}_{\oplus}^n$  as follows,

$$\mathbf{k} \sim p(\mathbf{k}) = \prod_{i=1}^n p(k_i) = \prod_{i=1}^n \frac{\pi_k^{k_i}}{k_i!} \times \exp\{-\pi_k\} = \frac{\pi_k^{\sum_i k_i}}{\prod_i k_i!} \times \exp\{-n\pi_k\},$$

which is a discretized spatially stationary Poisson count RF with  $\{\lambda(\mathbf{x}) = \lambda_k; \mathbf{x} \in D\}$  and  $\pi_k = \lambda_k \Delta_n$  and basis-event set  $\mathbb{E} = \emptyset$ .

c)

Now we want to estimate the intensity  $\lambda_k$  based on the observations with observation probabilities. To estimate the intensity, we start by estimating the event probability  $\pi_k$  in the discretized field represented by the  $n$ -vector  $\mathbf{k}$ . Hence, the model parameter for  $\mathbf{k}$  is  $\theta_{pP} = (\pi_k, \emptyset)$ . This is assessed through marginal maximum likelihood estimation,

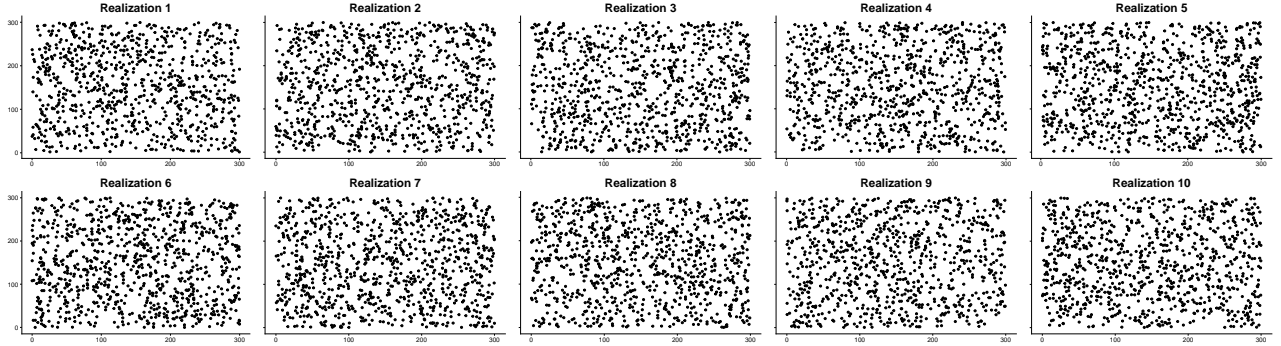
$$\hat{\theta}_{pP} = \underset{\theta_p}{\operatorname{argmax}} \{\log p(\mathbf{d}; \theta_p)\}.$$

By utilizing the joint density of  $\mathbf{k}$  and  $\mathbf{d}$ , and summing each term over  $k_i \geq d_i$ , we obtain the solution,

$$\hat{\pi}_k = \frac{n_d}{\boldsymbol{\alpha}^T \mathbf{i}_n},$$

where  $n_d$  is the number of event observations and  $\boldsymbol{\alpha}$  is the observation probabilities. Further, we get the estimated intensity  $\hat{\lambda}_k = \hat{\pi}_k / \Delta_n$ .

Having estimated the intensity, we generate 10 realizations from the prior Poisson event-count model and the associated approximate Poisson event-location realizations. The event-locations are then displayed.



d)

Having defined the likelihood model and the prior distribution, we now want to develop the expression for the posterior discretized event-count model. To do this, we first write the joint distribution for the spatial variable  $\mathbf{k}$  and the observations  $\mathbf{d}$ ,

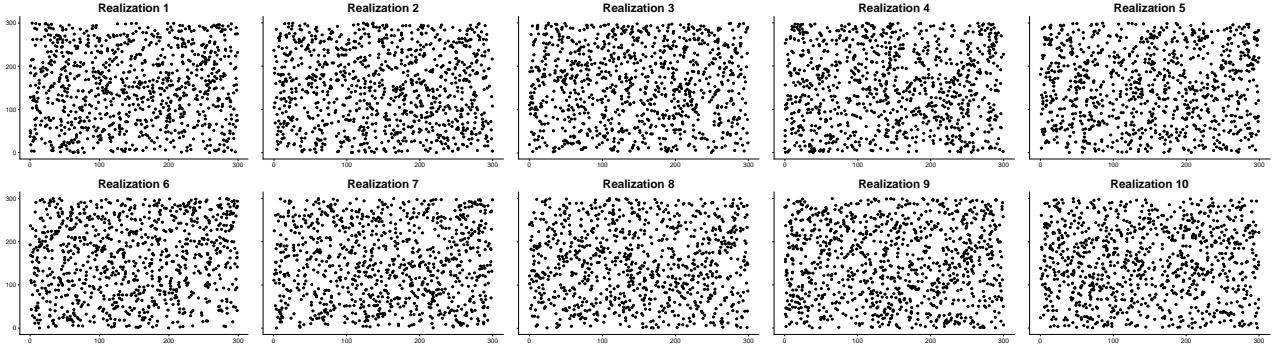
$$\begin{aligned} \begin{bmatrix} \mathbf{k} \\ \mathbf{d} \end{bmatrix} &\sim p(\mathbf{k}, \mathbf{d}) = p(\mathbf{d}|\mathbf{k})p(\mathbf{k}) = \prod_{i=1}^n p(k_i|d_i) \prod_{i=1}^n p(k_i) = \prod_{i=1}^n p(d_i, k_i) \\ &= \prod_{i=1}^n \binom{k_i}{d_i} \alpha_i^{d_i} [1 - \alpha_i]^{k_i - d_i} \times \frac{\pi_k^{k_i}}{k_i!} \exp\{-\pi_k\} \\ &= \prod_{i=1}^n \frac{[\alpha_i \pi_k]^{d_i}}{d_i!} \times \frac{[(1 - \alpha_i) \pi_k]^{k_i - d_i}}{(k_i - d_i)!} \times \exp\{-\pi_k\}. \end{aligned}$$

Having arrived at the joint distribution, we note that it is on factorial form, which implies independence of each grid node. Hence, the normalizing constant can be found by summing for each node over  $k_i \geq d_i$  to obtain  $p(d_i)$ . This means that the posterior distribution can be written as,

$$\begin{aligned} [\mathbf{k}|\mathbf{d}] &\sim p(\mathbf{k}|\mathbf{d}) = \prod_{i=1}^n [p(d_i)]^{-1} \times p(d_i, k_i) = \prod_{i=1}^n p(k_i|d_i) \\ &= \prod_{i=1}^n \frac{[(1 - \alpha_i)\pi_k]^{k_i - d_i}}{(k_i - d_i)!} \times \exp\{-(1 - \alpha_i)\pi_k\}, \end{aligned}$$

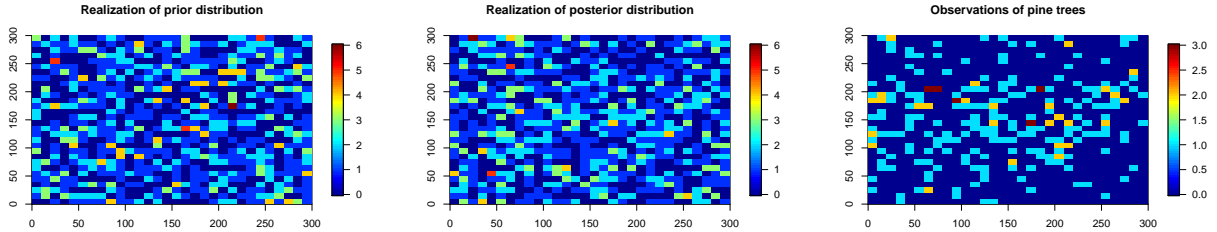
for  $\mathbf{k} \in \mathbb{N}_{\oplus}^n; \mathbf{k} \geq \mathbf{d}$ . This is a discretized Poisson count random field with model parameters  $\pi_n = ((1 - \alpha_1)\pi_k, (1 - \alpha_2)\pi_k, \dots, (1 - \alpha_n)\pi_k)$  and base event set  $\mathbf{k}_n^0 = (d_1, \dots, d_n)$ .

To simulate from the associated approximate event-location model, we use Algorithm 7 from (Omre 2019). Then we display 10 generated realizations from the posterior approximate event-location model.



If we compare the posterior approximate event-locations with the prior approximate event-locations, we see no clear difference. The posterior and prior realization look very similar. Still, in the posterior realizations we know that the observed points are present, so it will to some extent be different from the prior realizations, but this is hard to observe.

Finally we display three event-count plots. One for the prior model, one for the posterior and one for the observed points.



Here, the differences between the prior and posterior model are more visible. We see clearly that all counts in the observations are present in the posterior model, but not in the prior model, which stems from the fact that the posterior model has the observations as base set  $\mathbb{E}$ . Even though we see more clear differences in the event-count plots, they are still fairly similar in variability and overall appearance.

### Problem 3: Clustered event spatial variables

The Neuman-Scott cluster model (mother-child model) for event spatial variables is given below. The mother model is usually a stationary Poisson RF with intensity  $\lambda_M$ . Centered at every mother location,  $\mathbf{x}_j^M$ , a set of child events are independently located, with event count given by the pdf  $p(k^c)$  and intensity pdf  $p(\mathbf{x}|\mathbf{x}_j^M)$ . Censoring is done in order to make sure that the child-event is in  $D$ . The model can be expressed in terms of the event locations as  $\mathbb{X}_D^{NS} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{k_D})$ . The locations can also be expressed  $\mathbf{x}_{ji}$  with  $j = 1, \dots, k_D^M$ , and  $i = 1, \dots, k_{D_j}^c$  where  $k_D = \sum_j k_{D_j}^c$ , i. e. the  $i$ 'th child of mother  $j$ . We then obtain

$$\begin{aligned} \mathbb{X}_D^{NS} &\sim p(\mathbf{x}_{ji}; j = 1, \dots, k_D^M; i = 1, \dots, k_{D_j}^c) \\ &= \prod_{j=1}^{k_D^M} \left[ \prod_{i=1}^{k_{D_j}^c} \frac{I(\mathbf{x}_{ji} \in D) p(\mathbf{x}_{ji}|\mathbf{x}_j^M)}{P(D|\mathbf{x}_j^M)} \right] \times p(k_{D_j}^c | k_j^c, \mathbf{x}_j^M) p(k_j^c) \times \frac{1}{|D|^{k_D^M}} p(k_D^M) \\ &= \prod_{j=1}^{k_D^M} \left[ \prod_{i=1}^{k_{D_j}^c} \frac{I(\mathbf{x}_{ji} \in D) \phi_2(\mathbf{x}_{ji}; \mathbf{x}_j^M, \sigma_c^2 I_2)}{\Phi_2(D|\mathbf{x}_j^M, \sigma_c^2 I_2)} \right] \\ &\quad \times \binom{k_j^c}{k_{D_j}^c} \Phi_2(D|\mathbf{x}_j^M, \sigma_c^2 I_2) \left( 1 - \Phi_2(D|\mathbf{x}_j^M, \sigma_c^2 I_2) \right)^{k_j^c - k_{D_j}^c} p(k_j^c) \\ &\quad \times \frac{\lambda_M^{k_D^M}}{k_D^M!} \exp(-\lambda_M |D|), \end{aligned}$$

where  $k_{D_j}^j \leq k_j^c$ . The latter expression corresponds to the case of a mother-child model with a stationary Poisson RF as mother model, having intensity  $\lambda_M$ , and a Gaussian intensity child-model with spread  $\sigma_c^2$  where the child set size is given by  $p(k_j^c)$ . The parameters of this model are  $\theta_{pNS} = (\lambda_M, \sigma_c^2, p(\cdot))$ .

This model is not analytically tractable, but we can sample from it using the following algorithm.

```
simulate_neuman_scott = function(lambda_m, lambda_c, sigma_c) {
  # simulate one neuman scott process over (0, 1)x(-1, 0)

  result = matrix(NA, nrow = 0, ncol = 2)
  # total number of events
  k = 0
  # number of mothers
  k_m = rpois(1, lambda_m)
  # iterate through the mothers
  for (j in 1:k_m) {
    # sample uniformly from (0, 1)x(-1, 0) to get mother location
    x_m = c(runif(1, 0, 1), runif(1, -1, 0))
    # sample to get child count
    k_c = rpois(1, lambda_c)
    # place the children with mother as center
    for (i in 1:k_c) {
      x_c = mvrnorm(mu = x_m, Sigma = sigma_c * diag(2))
      # check if inside D
      if (x_c[1] > 0 && x_c[1] < 1 && x_c[2] > -1 && x_c[2] < 0) {
        result = rbind(result, x_c)
        k = k + 1
      }
    }
  }
}
```

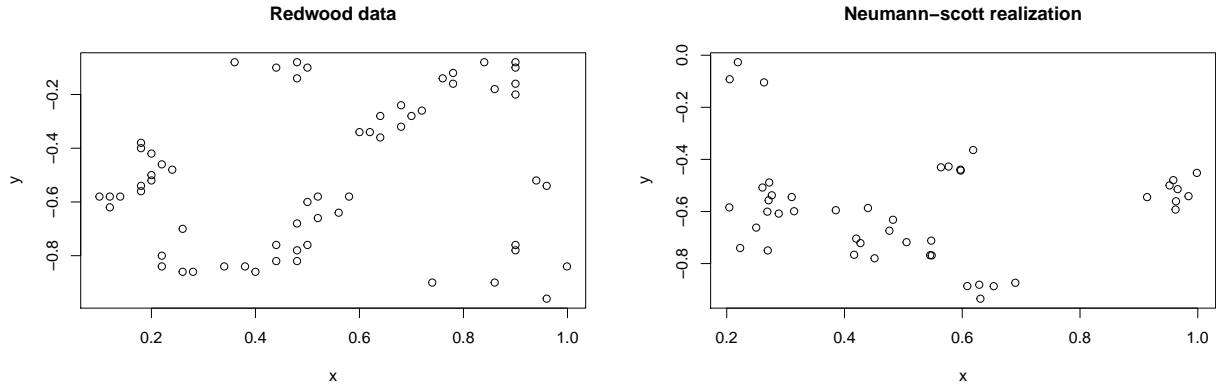
```

# returns a matrix where the rows are the observations
return(result)
}

```

This algorithm does however suffer from boundary bias, as we do not get any child contributions from potential mothers outside the domain  $D$ .

To get an empirical estimate of the model parameters, we study the redwood tree dataset. There seems to be approximately 10 clusters, and a total of 62 observations. Hence a mother intensity of 10 and a child intensity of  $62/10 = 6.2$  might be fitting. The spread in the clusters corresponds to a  $\sigma_c^2$  of approximately  $1.5 \cdot 10^{-3}$ , which we have obtained by trial and error. Below the redwood tree data is compared with one realization of this model.



The two plots look similar, but we want to justify our choice of parameters using an MCMC test on the  $L$ -function.

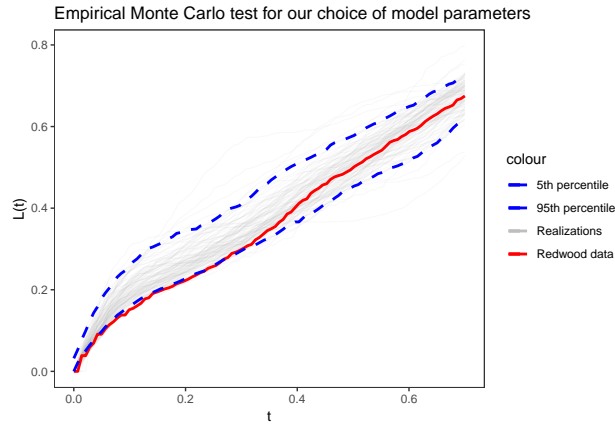


Figure 1: Empirical monte carlo test of our model parameters with mother intensity of 10 and child intensity of 6.2. We have computed the  $L$ -functions from 100 realizations of the Neumann-Scott model, and display these along with the 90% interval and the  $L$ -function of the Redwood data.

We note that the  $L$ -function is just outside our confidence interval for some small values of  $t$  for these parameters. But if we choose  $\lambda_M$  larger and  $\lambda_c$  smaller, we can get it completely contained in the confidence interval. The Empirical Montecarlo test for the Neumann-Scott model for  $\lambda_M = 15$  and  $\lambda_c = 3$  is plotted in figure 2.

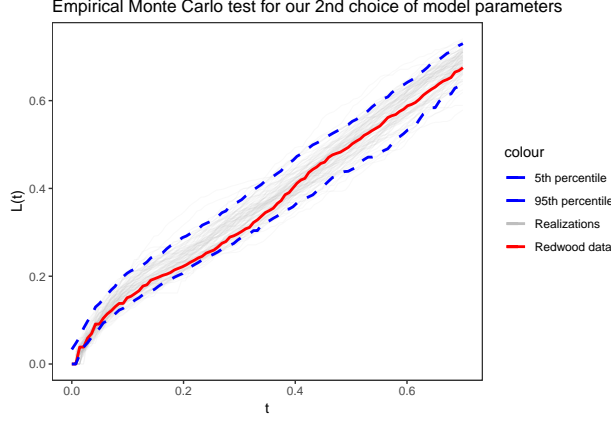
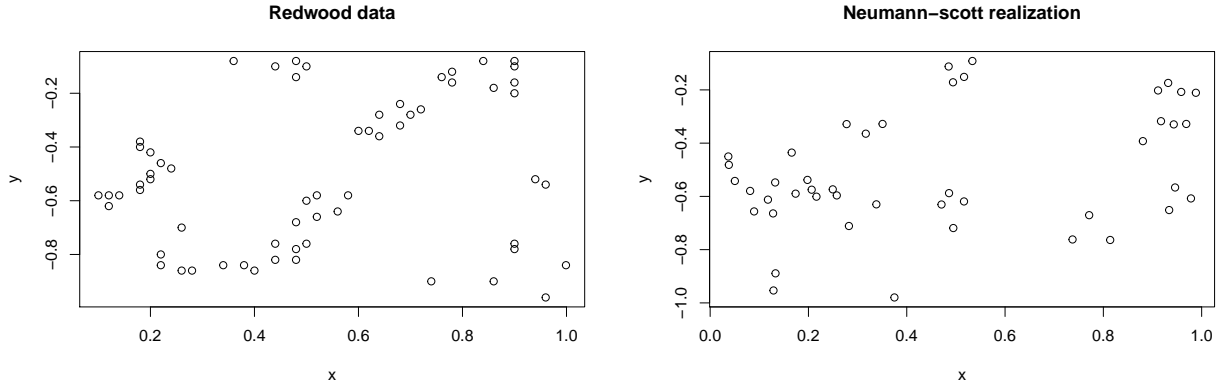


Figure 2: Empirical monte carlo test of our model parameters with mother intensity of 15 and child intensity of 3. We have computed the L-functions from 100 realizations of the Neumann-Scott model, and display these along with the 90% interval and the L-function of the Redwood data.

We can then display one realization and compare it to the Redwood data. This is done below.



From the displayed realization it is hard to tell if the parameters  $\lambda_M = 15, \lambda_c = 3$  fit the Redwood data better, even though the  $L$ -function of the Redwood data was contained in the 0.9-interval for this choice of parameters.

## Problem 4: Repulsive event spatial variables

The Strauss repulsion model for event spatial variables conditioned on the event count  $k_D$  is given below. The event locations  $\mathbb{X}_D^S$  given the event count is distributed as follows,

$$[\mathbb{X}_D^S | (k_D = k)] \sim p(\mathbf{x}_1, \dots, \mathbf{x}_k | k_D = k) \propto \prod_{i,j \in \{1, \dots, k\}} \exp(-\phi(\tau_{ij})),$$

with  $\tau_{ij} := \mathbf{x}_i - \mathbf{x}_j$  and  $\phi(\tau_{ij}) : \mathbb{R}^n \rightarrow \mathbb{R}_\oplus$  is decreasing with increasing  $|\tau_{ij}|$ .  $\phi(\cdot)$  is possibly anisotropic with  $\tau_{ij} = |\tau_{ij}| \in \mathbb{R}_\oplus$ . A typical form of  $\phi(\cdot)$  is

$$\phi(\tau) = \begin{cases} \phi_0, & \tau \in [0, \tau_0], \\ \phi_0 \exp(-\phi_1(\tau - \tau_0)), & \tau \in [\tau_0, \infty), \end{cases}$$



with model parameters  $\theta_{pS|k} = (\tau_0, \phi_0, \phi_1)$  all positive ( $\tau_0$  can be zero). This gives a repulsive model, as it enforces penalites (lower probability) whenever events are located close to eachother.

This Strauss event RF model is not analytically tractable, but we can simulate from it using the algorithm below.

```
# simulate one strauss repulsion process over (0, 1)x(0, 1)
simulate_strauss = function(k, tau0, phi0, phi1, iterations = 1000) {
  # we define the interaction function
  phi = function(tau) {
    dist = sqrt(sum(tau^2))
    if (dist <= tau0) {
      return(phi0)
    } else {
      return(phi0 * exp(-phi1 * (dist - tau0)))
    }
  }

  # initialize where initial value has nonzero probability we have
  # chosen to initialize the points uniformly on D
  result = matrix(runif(k), nrow = k, ncol = 2)

  # the algorithm is asymptotically correct. Increase iterations to
  # improve result
  for (i in 1:iterations) {
    # choose one index uniformly
    u = sample(1:k, 1)

    # generate a location uniformly
    x_p = runif(n = 2)

    # compute acceptance probability alpha

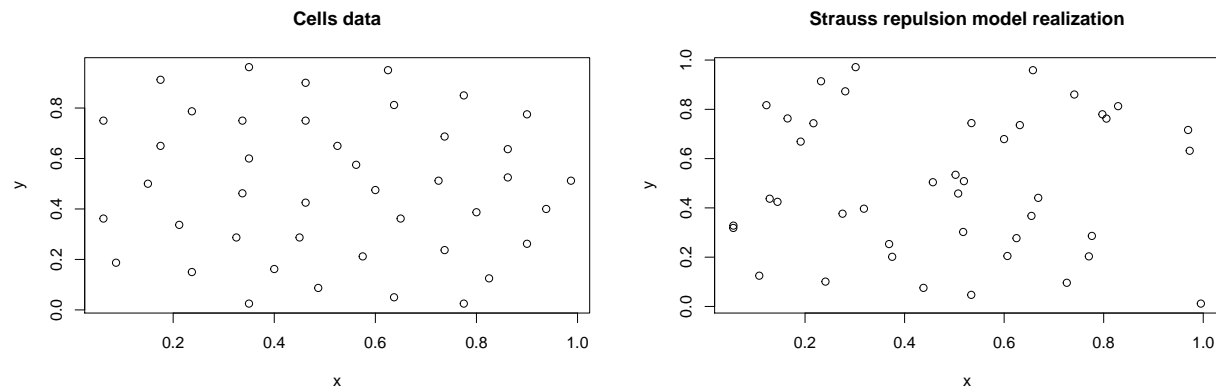
    alpha = 0
    for (j in 1:k) {
      alpha = alpha + phi(result[u, ] - result[j, ]) - phi(x_p -
        result[j, ])
    }
    alpha = exp(-alpha)
    alpha = min(1, alpha)

    # accept or reject
    if (runif(1) <= alpha) {
      # accept
      result[u, ] = x_p
    }
  }
  # returns a matrix where the rows are the observations
  return(result)
}
```

This algorithm also suffers from boundary bias, as we ignore possible contributions from events outside the boundary.

To get an empirical estimate of the model parameters, we study the biological cells dataset. There is a total of 42 observations. Hence we set  $k$  equal to 42. To obtain estimates of  $\tau_0, \phi_0$  and  $\phi_1$  one approach could be

to fix  $\tau_0$  and then find the ML estimates of  $\phi_0$  and  $\phi_1$ .  $\tau_0 = 0.1$  and  $\phi_0, \phi_1$  large would seem fitting - But the algorithm seems to converge slow for these values. So to get the simulations done in reasonable time we set  $\tau_0 = 10^{-5}$ ,  $\phi_0 = 10^{-3}$  and  $\phi_1 = 100$ . These parameters does not fit the data well. Iterating 1000 times seems to work for these values. Below the biological cells data is compared with one realization of this model.



We now want to study our choice of parameters using an MCMC test on the  $L$ -function. We have done this below.

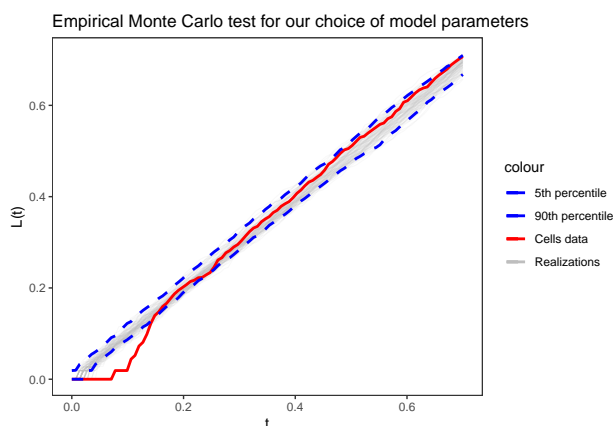


Figure 3: Empirical monte carlo test of our model parameters. We have computed the  $L$ -functions from 100 realizations of the Strauss repulsion model, and display these along with the 90% interval and the  $L$ -function of the Cells data.

The plot in figure 3 clearly shows that this choice of parameter values does not fit the data. Moreover, and a bit concerning, is that the  $L$ -function of the simulations lies above the line  $L(t) = t$ , which we expect for a clustering model, not a repulsive model.

## References

Omre, Henning. 2019. “Bayesian Spatial Inversion with Conjugate Prior Models,” January.