TMA4250 Spatial statistics - Project 2

Lars Evje, Martin Tufte

February 2022

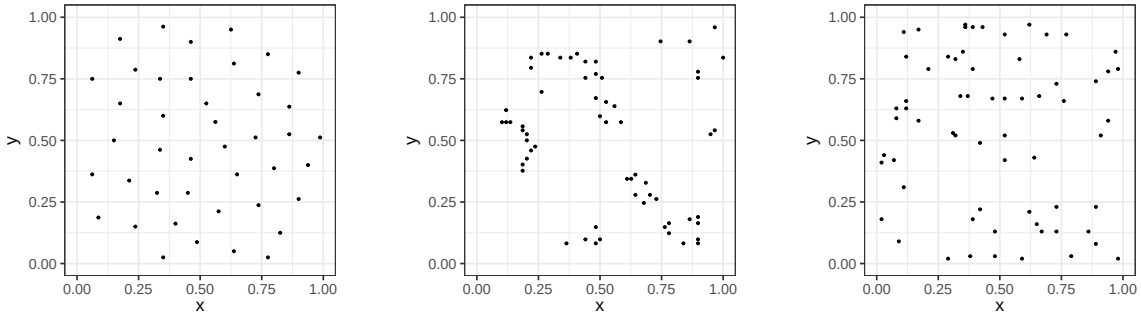
Problem 1: Analysis of point pattern data

a) In this problem we analyse three real-world data sets. Each data set consists of point pattern data in the area $B = [0,1]^2 \in \mathbb{R}^2$, and are displayed in Figure 1. The first is data from biological cells, the second is data over redwood trees in California and the third is data of Japanese pine trees.

From Figure 1a, it appears that the location of each cell is evenly spaced. The cells are forming one large cluster, but the distance between each pair of neighboring cells suggest that there is a repulsion between them. This is reasonable, as the cells are physical entities occupying separate spherical volumes of space.

Figure 1b displays a typical clustering behavior, as there are multiple redwood trees next to each other in separate regions of space. This is also reasonable, as some areas might be more suitable to grow the trees. Another contributing reason could be that the seeds from the redwood trees might not spread as far, limiting where new trees will grow up.

The data in 1c appears more random, where the pine trees form a large forest. This could for instance be due to little or no differences in the suitability of where pine trees can grow and that the seeds spread further.



(a) Biological cells data. (b) California redwood data. (c) Japanese pines data.

Figure 1: All three real-world point pattern data sets considered.

b) The empirical L-function for each of the point patterns can be calculated by using the build-in function `Kfn` in the R library `spatial`. The estimated L-functions are calculated using 100 regularly spaced distances for distances $r \in (0,\ 0.5)$ and displayed in Figure 2. A theoretical line for a homogeneous Poisson process is derived below and plotted as a reference.

Let N be a homogeneous Poisson point process over $\mathbb{R}^2$ with intensity λ. The theoretical L-function for N in $\mathbb{R}^2$ is given as

$$L(r) = \sqrt{\frac{K(r)}{\pi}}, \tag{1}$$

where $K(r)$ is Ripley's K-function

$$K(r) = \frac{1}{\lambda} \mathrm{E}_{\vec{0}} \left[N\left(b(\vec{0}, r) \setminus \{\vec{0}\} \right) \right]. \tag{2}$$

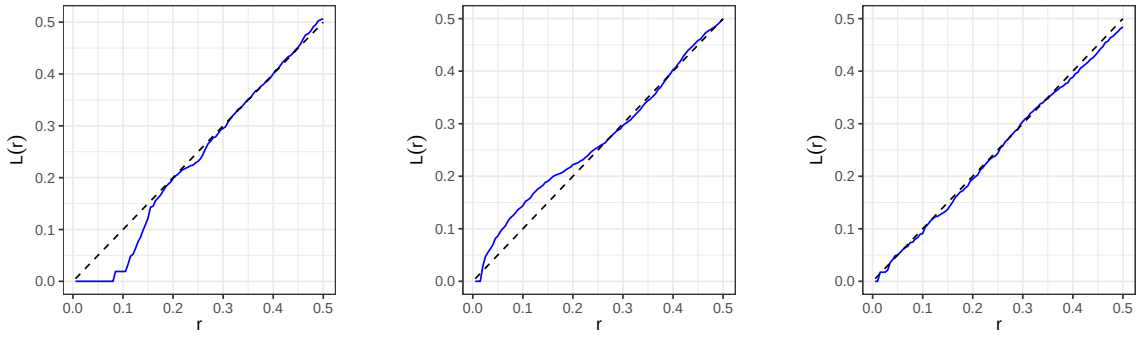
Note that the expectation in Equation (2) is assuming an observation in point $\vec{0} = (0, 0)$. However, since N is a Poisson process, $N(B) \sim \text{Poisson}(\lambda\nu(B))$, where $\nu(B)$ is the volume of $B \subset \mathbb{R}^2$. Since removing a single point $\{\vec{0}\}$ does not change the volume, it follows that

$$K(r) = \nu(b(\vec{0}, r) \setminus \{\vec{0}\}) = \pi r^2,$$

where we used that $\mathrm{E}[N(B)] = \lambda\nu(B)$. From Equation (1) we then get $L(r) = \sqrt{\frac{\pi r^2}{\pi}} = r$ for a homogeneous Poisson process.

For the biological cells point patterns in Figure 2a, there is a clear repulsion for $r < 0.15$. The redwood trees point patterns in Figure 2b have a clustering behavior for $r < 0.25$. For the pines tree point patterns in Figure 2c, there is no apparent deviation from the theoretical line produced from a Poisson point process.

A homogeneous Poisson RF appears to be a suitable model for the pines data. The same cannot be said about the cells data or the redwood data, as they both have large deviations from the theoretical line, displaying repulsive and clustering behaviour, respectively.

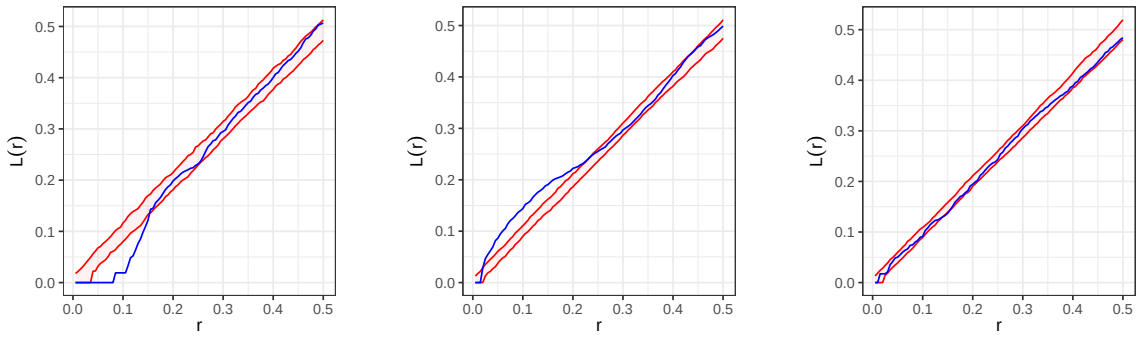


(a) Biological cells data. (b) California redwood data. (c) Japanese pines data.

Figure 2: Empirical $L(r)$ for the three data sets (blue line) and the theoretical $L(r) = r$ from a homogeneous Poisson point process (black dashed line).

c) Assuming that each data set arose from a homogeneous Poisson point process, a simulation can be made to create prediction intervals for the L-function. To simulate from the 2D Poisson distribution, an estimate of the intensity λ is needed. Using the unbiased estimator $\hat{\lambda} = \frac{n}{\nu(B)}$, where there are n observations in the area B, we have that $\nu(B) = \nu([0,1]^2) = 1$, so that the estimator for the intensity is simply $\hat{\lambda} = n$, i.e. the number of points.

100 realizations are simulated for each data set assuming the processes arose from a homogeneous Poisson process. A 90% prediction interval along with the empirical estimate of $L(r)$ is displayed in figure 3. Similarly as found in section b), the cells and redwood data seems to not have been produced by a homogeneous Poisson process. The empirical L-function for the pines data lies inside the prediction region, suggesting that a homogeneous Poisson RF appears to be a reasonable model.

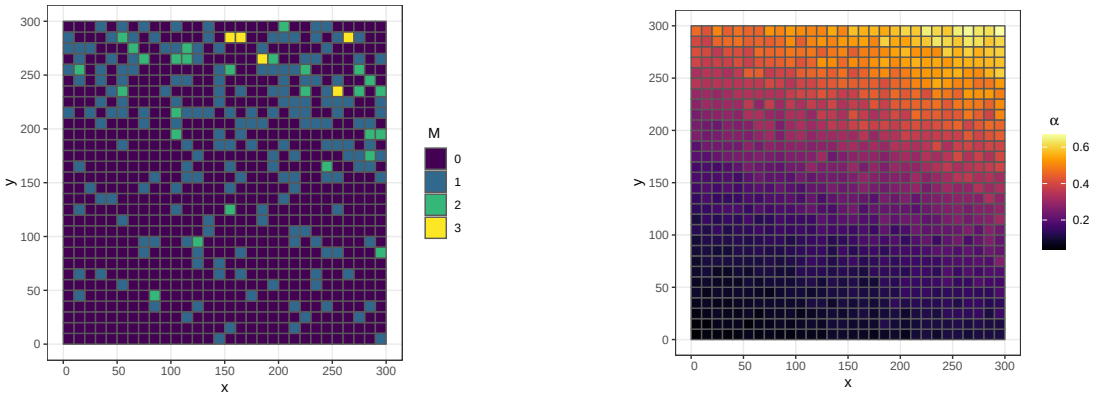


(a) Biological cells data. (b) California redwood data. (c) Japanese pines data.

Figure 3: Empirical $L(r)$ for the three data sets (blue line) and 90% prediction intervals assuming a homogeneous Poisson point process from 100 simulations (red region).

Problem 2: Remote sensing of trees

We will now look at a 300 m $\times$ 300 m observation window in a pine tree forest. A satellite has provided data consisting of counts of pine trees inside 10 m $\times$ 10 m grid cells for a regular 30×30 grid of the observation window. There is a detection probability for individual trees varying across the window. We denote the true number of pines by $N_{i,j}$, detected number of pines by $M_{i,j}$ and the detection probability by $\alpha_{i,j}$, for grid cells $i, j = 1, ..., 30$. Let $\boldsymbol{N} = (N_{1,1}, ..., N_{30,1}, ...N_{30,30})^\top$, $\boldsymbol{M} = (M_{1,1}, ..., M_{30,1}, ...M_{30,30})^\top$, and $\boldsymbol{\alpha} = (\alpha_{1,1}, ..., \alpha_{30,1}, ...\alpha_{30,30})^\top$. We are given $\boldsymbol{M}$ and $\boldsymbol{\alpha}$.



(a) Counts of detected pine trees.

(b) Observation probabilities for the pine trees.

Figure 4: The number of pine trees counted in each gridcell and the detection probability of each cell.

a) Figure 4 shows the counts of detected pine trees and the observation probabilities. We note that the detection probability seems to increase as you go upwards and to the right, also the amount of observed pine trees follow the same trend. The number of detected pine trees in one gridcell is independent of all other gridcells. For each gridcell there are $N_{i,j}$ actual pine trees, so we get $N_{i,j}$ trials with a probability of $\alpha_{i,j}$ of successfully noticing the tree. Assuming each trial is independent, it follows that $M_{i,j}|N_{i,j} \sim Binom(N_{i,j}, \alpha_{i,j})$. Independence between gridcells gives

$$f(\boldsymbol{M}|\boldsymbol{N}) = \prod_{i,j} \binom{N_{i,j}}{M_{i,j}} \alpha_{i,j}^{M_{i,j}} (1 - \alpha_{i,j})^{N_{i,j} - M_{i,j}}.$$

b) We assume *a priori* that the pine trees follow a homogeneous Poisson point process with intensity λ. Since the different gridcells are disjoint (only intersect on sets of measure zero), then $N_{i,j} \overset{iid}{\sim} Poisson(\lambda A)$, where $A = 100$ is the area of a gridcell. Joint distribution is thus,

$$f(\boldsymbol{N}) = \prod_{i,j} \frac{(\lambda A)^{N_{i,j}}}{N_{i,j}!} e^{-\lambda A}.$$

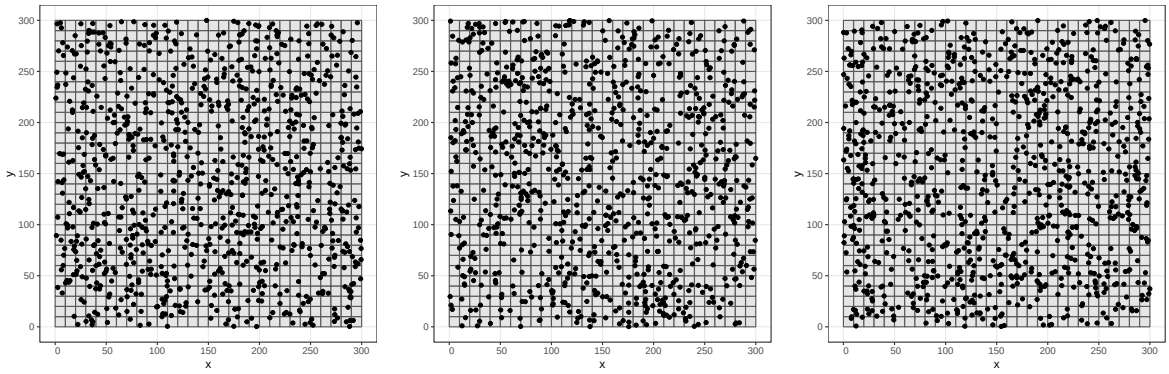
c) We seek to determine an unbiased estimator of λ of the form $\hat{\Lambda} = C \cdot \sum_{i,j} M_{i,j}$. Taking expectations we find

$$\begin{aligned} E[\hat{\Lambda}] &= E[C \cdot \sum_{i,j} M_{i,j}] = C \sum_{i,j} E[M_{i,j}] \\ &= C \sum_{i,j} E[E[M_{i,j}|N_{i,j}]] = C \sum_{i,j} E[N_{i,j}\alpha_{i,j}] \\ &= C \sum_{i,j} \lambda A \alpha_{i,j}, \end{aligned}$$

where we have used the law of total expectations. For $\hat{\Lambda}$ to be unbiased we must have $C = \frac{1}{A \sum_{i,j} \alpha_{i,j}}$, and our estimator becomes

$$\hat{\Lambda} = \frac{1}{A \sum_{i,j} \alpha_{i,j}} \sum_{i,j} M_{i,j}.$$

Using our estimator $\hat{\Lambda}$ on our dataset, we get $\hat{\lambda} = 0.0104$. Figure 5 shows simulated point patterns from three realizations of $\boldsymbol{N}$. As expected from a homogeneous Poisson process there seems to be no pattern. Comparing to figure 4, the realizations for $\boldsymbol{N}$ seems reasonable, since if N is homogeneous distributed across the domain the deciding factor for the number of observed trees are α. And we see, as mentioned in 1a), that the observed trees seems to correspond to values of α.



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 5: Three simulations of point patterns from realizations of $\boldsymbol{N}$.

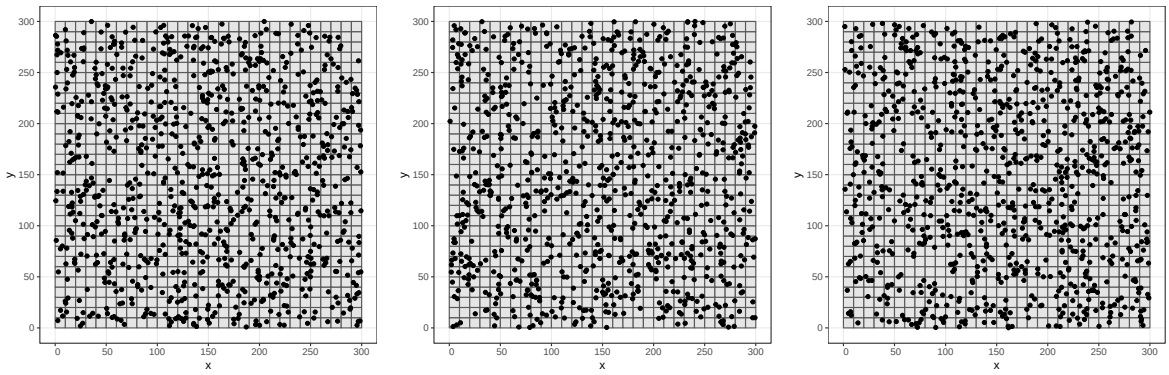
d) Next we want to derive the probability mass function of $\boldsymbol{N}|\boldsymbol{M} = \boldsymbol{m}$. Using the fact that everything is independent between gridcells we look at one gridcell and drop the indices:

$$
\begin{aligned}
f(N|M=m) = \frac{f(M=m|N)f(N)}{f(M=m)} &= \frac{\binom{N}{m}\alpha^m(1-\alpha)^{N-m}\frac{(\lambda A)^N}{N!}e^{-\lambda A}}{\sum_{n=0}^{\infty}\binom{n}{m}\alpha^m(1-\alpha)^{n-m}\frac{(\lambda A)^n}{n!}e^{-\lambda A}} \\
&= \frac{\frac{N!}{m!(N-m)!}(1-\alpha)^{N-m}\frac{(\lambda A)^N}{N!}}{\sum_{n=0}^{\infty}\frac{n!}{m!(n-m)!}(1-\alpha)^{n-m}\frac{(\lambda A)^(n}{n!}} = \frac{(1-\alpha)^{N-m}\frac{(\lambda A)^{N-m}}{(N-m)!}}{\sum_{n=0}^{\infty}(1-\alpha)^{n-m}\frac{(\lambda A)^{n-m}}{(n-m)!}} \\
&= \frac{\frac{N!}{m!(N-m)!}(1-\alpha)^{N-m}\frac{(\lambda A)^N}{N!}}{\sum_{n=0}^{\infty}\frac{n!}{m!(n-m)!}(1-\alpha)^{n-m}\frac{(\lambda A)^(n}{n!}} = \frac{(1-\alpha)^{N-m}\frac{(\lambda A)^{N-m}}{(N-m)!}}{\sum_{n-m=0}^{\infty}(1-\alpha)^{n-m}\frac{(\lambda A)^{n-m}}{(n-m)!}} \\
&= \frac{((1-\alpha)\lambda A)^{N-m}}{(N-m)!}e^{-(1-\alpha)\lambda A}.
\end{aligned}
$$

And so the mass function for $\boldsymbol{N}$ is

$$
f(\boldsymbol{N}|\boldsymbol{M}=\boldsymbol{m}) = \prod_{i,j}\frac{((1-\alpha_{i,j})\lambda A)^{N_{i,j}-m_{i,j}}}{(N_{i,j}-m_{i,j})!}e^{-(1-\alpha_{i,j})\lambda A}.
$$

We recognize this as the probability mass function of an Poisson distribution for $N_{i,j}-m_{i,j}$ with parameter $(1-\alpha)\lambda A$. This indicates that the point process of number of unobserved points, $\boldsymbol{N}-\boldsymbol{M}$ is an inhomogeneous Poisson point process with intensity $(1-\alpha)\lambda$, where $\alpha = \alpha(x,y)$. Figure 6 shows simulated point patterns from three realizations of $\boldsymbol{N}|\boldsymbol{M}=\boldsymbol{m}$. These realizations look very similar to the realizations in Figure 5, which might indicate that our *a priori* assumption for $\boldsymbol{N}$ was quite good. It is however hard to judge from these plots, whether there is any difference in intensity between the two (if amount of points are similar).

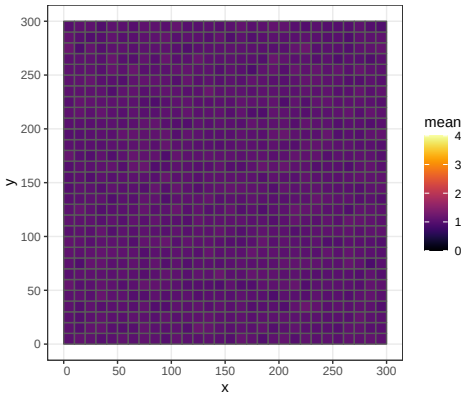


(a) Realization 1. (b) Realization 2. (c) Realization 3.

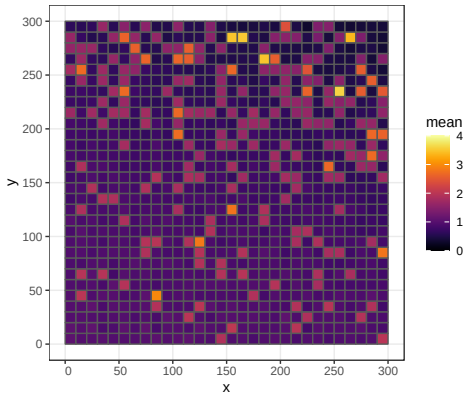
Figure 6: Three simulations of point patterns from realizations of $\boldsymbol{N}|\boldsymbol{M}=\boldsymbol{m}$.

e) Figure 7 shows the estimation of the expected value and standard deviations of $\boldsymbol{N}$ and $\boldsymbol{N}|\boldsymbol{M}=\boldsymbol{m}$, based on 500 realizations. The expected value of $\boldsymbol{N}$ seems to be close to the value we would expect (based on our estimate of λ) $\hat{\lambda}\cdot A = 1.04$, and seems to be homogeneous. The estimate for

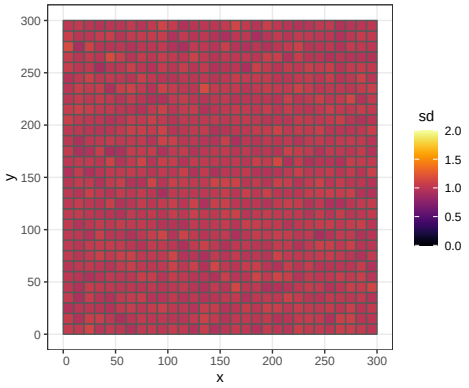
$E[\boldsymbol{N}|\boldsymbol{M} = \boldsymbol{m}]$ seems to give higher values overall, and they seem to increase for larger x and y values. Seems to indicate that maybe an inhomogeneous process would be more appropriate. The standard deviation for $\boldsymbol{N}$, is close to what one would expect (i.e. 1) and seems homogeneous. The standard deviation of $\boldsymbol{N}|\boldsymbol{M} = \boldsymbol{m}$ seems to be quite similar except in the upper right corner where it is lower. This makes sense since the detection probability was largest there, and one would expect to be closer to the true value.



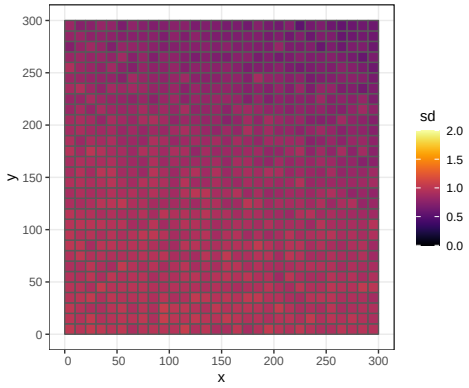
(a) Estimate of $E[\boldsymbol{N}]$.



(b) Estimate of $E[\boldsymbol{N}|\boldsymbol{M} = \boldsymbol{m}]$.



(c) Estimate of $SD[\boldsymbol{N}]$.



(d) Estimate of $SD[\boldsymbol{N}|\boldsymbol{M} = \boldsymbol{m}]$.

Figure 7: Estimates of expected values and standard deviations of $\boldsymbol{N}$ and $\boldsymbol{N}|\boldsymbol{M} = \boldsymbol{m}$.

Problem 3: Clustered event spatial variables

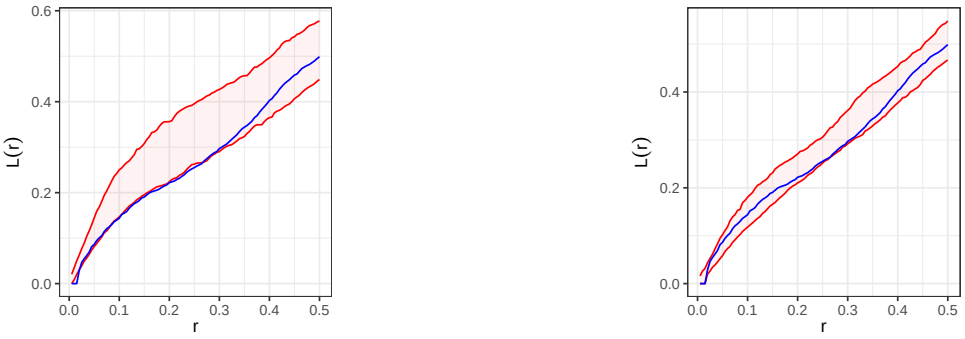
This problem concerns the redwood tree data set.

a) We consider a Neyman-Scott process, where the number of daughter points of a parent is distributed according to a Poisson distribution, and locations of daughter points are generated according to $\mathcal{N}_2(\boldsymbol{y}; \sigma^2 \mathbf{I}_2)$, where $\boldsymbol{y}$ is the parent location. The generation of daughter points can be summarized in two steps:

1. Generate parent points $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots\}$ according to a homogeneous Poisson process with intensity $\lambda_p > 0$.
2. For each parent point $\boldsymbol{y} \in \mathbb{R}^2$, we generate the number of daughter points n_d according to $n_d \sim \text{Poisson}(n_d; \lambda_d)$. Generate daughter points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots\}$ independently by $\boldsymbol{x}_i = \boldsymbol{y} + \boldsymbol{r}_i$, where $\boldsymbol{r_i} \sim \mathcal{N}_2(\mathbf{0}, \sigma^2 \mathbf{I}_2)$ is the displacement vector from the parent location.

The full set of model parameters are $\theta = \{\lambda_p, \lambda_d, \sigma^2\}$. The parameter λ_p is the intensity of the number of parent points, and equals the average parent points in a unit volume. The parameter λ_d is the average number of daughter points per parent. This means that the overall intensity of daughter points in the full model is equal to $\lambda_p \lambda_d$, i.e. the expected number of daughter points multiplied with the intensity of parent points. The parameter σ^2 is the variance of the distribution of daughter points around a parent point. If σ^2 is small, the daughter points will be close to the location of the parent.

When simulating realizations from the model we need to restrict the domain of interest. Assume we want to sample from a finite domain $W \subset \mathbb{R}^2$. Then, since the locations of the daughter points are picked at random centered around the parent points, there could be some parent point $\boldsymbol{y} \notin W$, but that has daughter points inside the boundary of W. If we restrict the generation of parent points to be inside W, then this border problem can arise. To address this issue, we generate parent nodes in $W_{\pm\Delta} = [-\Delta, 1 + \Delta]^2 \subset \mathbb{R}^2$ for a value of $\Delta \geq 0$.



(a) Using initial guessed parameters.

(b) Using iterated parameters.

Figure 8: Empirical $L(r)$ for the redwood trees data sets (blue line) and 90% prediction intervals assuming a Neymann-Scott process from 100 simulations (red region).

An empirical fit of the parameters in θ can be found from inspecting Figure 1b. There are 62

observations in total and it appears to be about 8 clusters. This gives an empirical estimate of $\lambda_p \approx 8$ and $\lambda_d \approx 62/\lambda_p = 7.75$. From a visual inspection we estimate that most of the points in one cluster is located within a circle of radius $r \approx 0.1$. This means that the standard deviation is roughly 0.05, and hence $\sigma^2 \approx 0.05^2 = 0.0025$. An evaluation of this guess is made by simulating the Neymann-Scott process 100 times. To correct for boundary effects when simulating we used $\Delta = 0.25$. A 90% prediction interval of the empirical L-function under the chosen parameter values are displayed in Figure 8. We see that the true L-function lies within the prediction interval produced by our initial guess. However, there are room for improvement.

We seek to optimize θ iteratively to find the parameters that most likely produced the observed data assuming the data arose from a Neymann-Scott process. To evaluate the fit, we used the integrated squared difference as our objective, given by

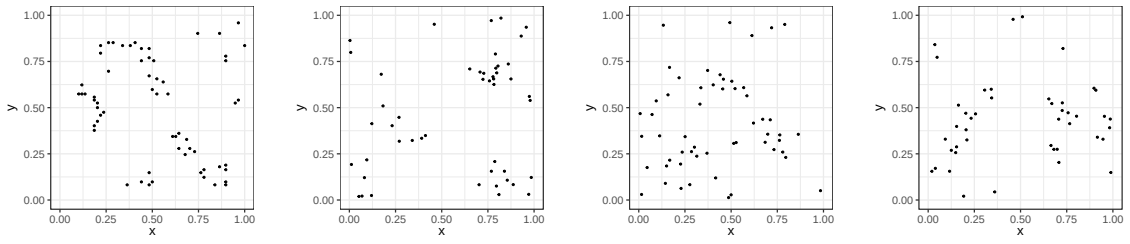
$$\mathcal{L}(\theta) := \int_0^{\frac{1}{2}} \left(L(r) - \mathrm{E}[L_{NS}(r) \mid \theta]\right)^2 \mathrm{d}r, \tag{3}$$

where $L(r)$ is the empirical function and $\mathrm{E}[L_{NS}(r) \mid \theta]$ is the estimated $L(r)$ given the parameter values θ. The iterations was preformed starting with our initial guess, denoted by θ_0. Then we picked 20 new parameter vectors $\theta_{1,1}, \theta_{1,2}, \cdots, \theta_{1,20}$ with small perturbations of each parameter. The new parameters are able to change by upwards of 10% per iteration, but are restricted such that $\lambda_p \lambda_d = 62$. Then we pick the next iterate as

$$\theta_k = \underset{\theta_{k-1,i}}{\operatorname{argmin}}\ \mathcal{L}(\theta_{k-1,i}),$$

where $\mathcal{L}(\theta_{k-1,i})$ is calculated numerically. After 8 iterations our new estimates for the parameters became $\lambda_p = 14.15$, $\lambda_d = 4.38$ and $\sigma^2 = 0.0030$. The new estimate for $L(r)$ is displayed in Figure 8b. We see that our model fits well, but that the true data appear to be more repulsive at around $r = 0.3$.

To check that the model is reasonable, three simulations using the new parameters are displayed with the true data set in Figure 9. Although the simulations seems to resemble the data, there seems to be more structure in the redwood point pattern data, with the points lining up more, suggesting that the data is not completely isotropic.



(a) Redwood data. (b) Simulation 1. (c) Simulation 2. (d) Simulation 3.

Figure 9: California redwood data (a) and three simulations assuming a Neymann-Scott process with optimized parameters (b) - (d).

Problem 4: Repulsive point processes

This problem concerns the biological cell data set.

a) We model the point pattern using a Strauss process with a fixed number of points n and pair potential function given by

$$\phi(r) = \begin{cases} \beta, & r \leq r_0 \\ 0, & r > r_0 \end{cases}. \tag{4}$$

Equation (4) describes the potential energy caused by interaction between a pair of points $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as $\phi_{ij} = \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$. The total potential energy in the system $u(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ is found from summing up all the contributions from the pairs, giving

$$u(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) = \sum_{j>i} \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \tag{5}$$

where the notation $\sum_{j>i} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n}$. Restricting our domain to $W \subset \mathbb{R}^2$ and assuming a fixed number of points n, we have a Gibbs process. The probability density of the state with $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in W$ is then given as

$$f\left(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \mid N(W) = n\right) = \frac{1}{Z_n} \exp\left(-u(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)\right), \tag{6}$$

where Z_n is a normalization constant. This normalization constant cannot be calculated numerically as there are an infinite amount of possible states in the continuous domain W.

The full set of model parameters are $\theta = \{\lambda, \beta, r_0\}$. The parameter λ is the intensity of the points, and equals the average number of points in a unit volume. The parameter β is the potential energy between a pair of points located within a proximity $r \leq r_0$. If β is high, there is a large penalty for picking a state where two points are closer to each other than r_0.

There are a couple of potential problems to consider with this model. The first is a border problem related to points outside W which could influence the potential energies of points inside of W. This means that if there are points outside of W within a radius of r_0 of the border, there could be potential points in W which should have been influenced by the outside point. The second is that we restrict the number of cells inside of W to a fixed number. In reality the cells are moving entities and could cross the border of W. We ignore both of these problems in our implementation.

To sample from this distribution, we use an iterative MCMC where a new proposal state is generated by updating the location of each point one by one. Let $\vec{\boldsymbol{x}} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ be the current state, and $\vec{\boldsymbol{x}}' = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{k-1}, \boldsymbol{x}'_k, \boldsymbol{x}_{k+1}, \cdots, \boldsymbol{x}_n)$ be the new proposal where $\boldsymbol{x}'_k$ is picked uniformly from $[0,1]^2$. The acceptance probability becomes

$$\alpha(\vec{\boldsymbol{x}}, \vec{\boldsymbol{x}}') = \min\left\{1, \frac{f\left(\vec{\boldsymbol{x}}' \mid N(W) = n\right)}{f\left(\vec{\boldsymbol{x}} \mid N(W) = n\right)}\right\} = \min\left\{1, \exp\left(u(\vec{\boldsymbol{x}}) - u(\vec{\boldsymbol{x}}')\right)\right\},$$

which simplifies to

$$\alpha(\vec{\boldsymbol{x}}, \vec{\boldsymbol{x}}') = \min\left\{1, \exp\left(\sum_{j \neq k} (\phi(\|\boldsymbol{x}_k - \boldsymbol{x}_j\|) - \phi(\|\boldsymbol{x}'_k - \boldsymbol{x}_j\|))\right)\right\}.$$

The simulations are done by first initializing a random state consisting of the locations of the 42 points and iterating until the energy calculated using Equation (5) indicates convergence.

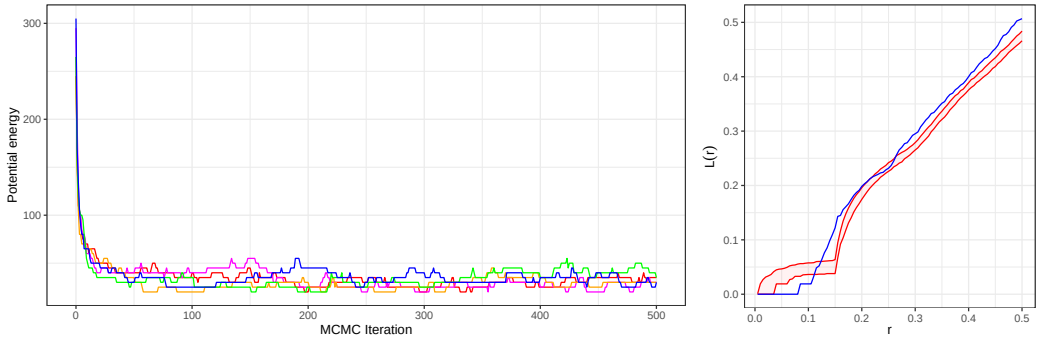
As in problem 3, we make an initial guess of the parameters of θ. Since there are 42 cells in the data set, we set $\lambda = 42$ and hence use $n = 42$ cells in all our simulations. From inspecting Figure 1a and 2a, it appears that the cells do not come any closer than about $r \approx 0.15$. We therefore set our initial guess as $r_0 = 0.15$. The parameter β is harder to guess, but it has to be greater than one for it to be a Strauss process, and not only a homogeneous Poisson process. Since the repulsion seems to be high, let $\beta = 5$ for now. To ensure that each simulation converges, the MCMC sampler needs to do enough iterations for each simulation. Plotting the energy as a function of the iteration number, we see from Figure 10a that the sampler converges after about 100 iterations. We therefore use 100 iterations of our implementation when sampling from the Strauss process.

The estimated $L(r)$ using the guessed parameters are displayed in Figure 10b. We see that there is a clear repulsion for $r < 0.15$, but the shape does not quite line up with the empirical $L(r)$. Using the same procedure as in problem 3 to iterate the parameters, with 5 new proposals in each iteration, we got the parameters $\beta = 4.15$ and $r_0 = 0.11$ after 10 iterations. The new estimated $L(r)$ with the iterated parameters are displayed in Figure 11b. We see that the empirical $L(r)$ lies almost consistent with the 90% prediction regions using the iterated parameters.

This model does not fit the data perfectly. In Figure 11b, we see that for $r < 0.08$, the estimated $L(r)$ does not equal zero as the empirical graph. A way to change the model such that the fit for small values of r is better is to use a different potential function $\phi(r)$. One might change to a hard-core process for a $r_1 < r_0$, such that the potential becomes on the form

$$\phi(r) = \begin{cases} \infty, & r < r_1 \\ \beta, & r_1 \leq r \leq r_0 \\ 0, & r > r_0 \end{cases}.$$

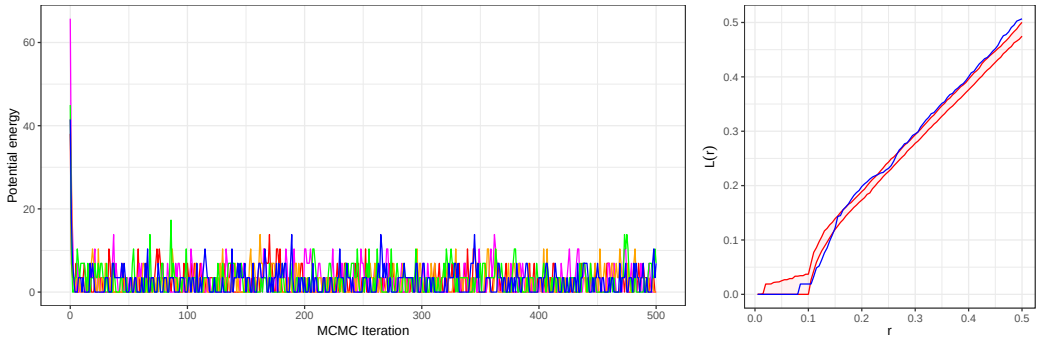
Another way to change the potential function could be to use a logarithmic barrier function.



(a) Potential energy.

(b) Estimated $L(r)$.

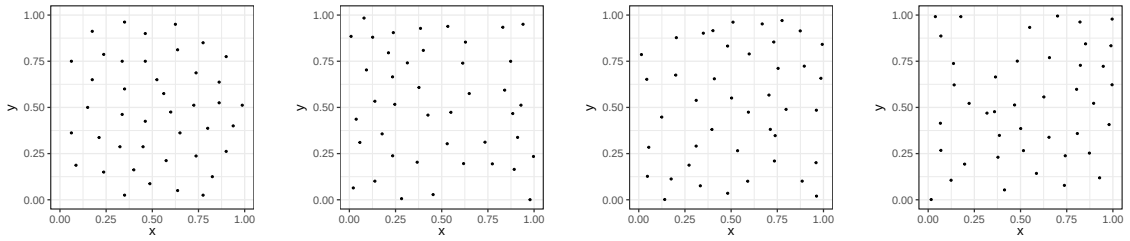
Figure 10: Using the guessed parameters to sample from the Strauss process. (a) shows is the potential energy for five samples in the MCMC sampler. (b) shows the empirical $L(r)$ along with a 90% prediction interval using 100 simulations (red region).



(a) Potential energy. (b) Estimated $L(r)$.

Figure 11: Using the iterated parameters to sample from the Strauss process. (a) shows is the potential energy for five samples in the MCMC sampler. (b) shows the empirical $L(r)$ along with a 90% prediction interval using 100 simulations (red region).

To check that the model is reasonable, three simulations using the new parameters are displayed with the true data set in Figure 12. Although the simulations seems to resemble the data, there are some points in the simulations which are very close to each other. Another interesting find is that the true data set has no cells located in the corners, while our simulation does. The simulations also seem to have large areas in-between the cells, which seems unnatural in regards to the true data.



(a) Cells data. (b) Simulation 1. (c) Simulation 2. (d) Simulation 3.

Figure 12: Biological cells data (a) and three simulations assuming a Strauss process with optimized parameters (b) - (d).