The Journal of Open Source Software

The ppmData R-package for setting up spatial point process models

Skipton N. C. Woolley [1,2] **and Scott D. Foster** [3]

1 Environment, CSIRO **2** School of Ecosystems and Forestry Science, The University of Melbourne **3** Data61, CSIRO

DOI: 10.21105/joss.04771

Software

- Review
- Repository
- Archive

Editor: Jayaram Hariharan

Reviewers:

- @OwenWard
- @mhesselbarth

Submitted: 03 August 2022
Published: 27 February 2023

Summary

Spatial data, the locations of objects in a spatial region, is a common type of data recorded in ecology, environmental sciences and epidemiology amongst many others. An appropriate statistical approach for data which can be described as points, such as plants or individuals of an animal population, are spatial point process models (Baddeley & Turner, 2000; Cressie, 1993). Spatial point processes can be used to understand the location of objects to inform decision making and improve ecological, health and other socio-economic outcomes. For example, the locations of individuals from a species' population are typically not distributed completely at random. Locations might be associated with favorable habitat conditions for the niche of that species. We might expect these species locations and the underlying spatial intensity are distributed according to relevant covariates, such as environmental conditions at those locations.

Inhomogeneous Poisson Process Models allow for the spatial intensity of a point process to vary across space, usually as a function of spatially referenced covariates. One challenge when fitting an Inhomogeneous Poisson Process Model is that the integral used to described the intensity surface in the log-likelihood (as described below) has no known closed analytic solution. For Inhomogeneous Poisson Process Models, the Berman-Turner device (Berman & Turner, 1992) is commonly used to approximate the model using quadrature within a weighted Poisson generalized linear model. The type of quadrature scheme can have important ramifications for overall model fitting and inference (Warton & Shepherd, 2010). Approximating the integral accurately often comes at an increased computation cost, where a large number of quadrature points are required to robustly estimate the integral (Renner et al., 2015).

Our `ppmData` package is setup to use a quasi-random quadrature approach as an alternative to pseudo-random sampling (Phillips et al., 2009) and regular grid (Warton & Shepherd, 2010) quadrature approaches. Quasi-random sampling, a base-case of Balanced Acceptance Sampling (Robertson, Brown, McDonald, & Jaksons, 2013), can be used to efficiently perform numerical integration (Halton, 1960) and create spatially balanced survey designs. Quasi-random sampling is an efficient form of spatial sampling as it approximately balances over all spatially smooth covariates (Grafström & Tillé, 2013), even if they are not included or considered in the quadrature generation. When used in the fitting of a Inhomogeneous Poisson Process Models, quasi-random (spatially balanced) quadrature also reduces the influence of spatial autocorrelation between quadrature points and is likely to improve numerical approximation of the intensity (Foster et al., 2017; Liu & Vanhatalo, 2020).

Here we present the `ppmData` R package that is designed to setup a quadrature scheme using Dirichlet tessellation for fitting spatial point process models. `ppmData` can setup a quadrature scheme of point process (e.g the locations of species) or a marked point process (e.g the locations of multiple species, where each location is associated with a specific species). In this paper, we demonstrate how to set up quadrature for inhomogeneous Poisson Process Models

and provide a simple example of how we can fit an inhomogeneous Poisson Process Model using a `ppmData` object.

Inhomogeneous Poisson Process Models

An Inhomogeneous Poisson Process Model is a statistical model used to describe a random point pattern for n observed locations, $(y = (y(s_1), y(s_2), \dots, y(s_n))$, in a spatial window $\mathcal{A} \subset R^2$. An Inhomogeneous Poisson Process Model assumes that the number of points in $\mathcal{A}$, n, is a Poisson random variable with total intensity Λ defined as $\Lambda = \int_{s \in \mathcal{A}} \lambda(s) ds$. The spatial function $\lambda(s)$ is usually a non-constant function and defines the spatial surface that the spatial locations y are independently drawn from (Cressie, 1993). For convenience, we use subscript notation for particular spatial locations, for example, $\lambda_i = \lambda(s_i)$ is the intensity surface of the point process at the i^{th} location with coordinates s_i. The intensity surface can be defined as $\log\{\lambda_i\} = x_i^\top \beta$ where x_i is a vector of spatially located covariates and β is a corresponding vector of coefficients.

The approximate log-likelihood $\ell(\beta; \{x_i\})$ can be written as a weighted Poisson likelihood (Berman & Turner, 1992)

$$\begin{aligned} \ell(\beta|y) &= \sum_{i=1}^{n} \log(\lambda_i) - \int_{s \in \mathcal{A}} \lambda(s) ds - \log(n!) \\ &\approx \sum_{i=1}^{n} \log(\lambda_i) - \sum_{i=1}^{m} w_i \lambda_i \\ &= \sum_{i=1}^{m} w_i \{z_i \log(\lambda_i) - \lambda_i\} \end{aligned} \tag{1}$$

where z_i is an indicator variable identifying if site i is one of the n observed points from the random point pattern or one of the m quadrature points. w_i stores the quadrature weights which sum to the total area of the region $|\mathcal{A}|$. In (1), we can see the integral in the log-likelihood $\ell(\beta|\mathrm{y})$ is being estimated using quadrature (Baddeley & Turner, 2000; Diggle, Menezes, & Su, 2010). Warton & Shepherd (2010) demonstrated for presence-only species distribution models, that using numerical quadrature with a point process framework makes the models scale invariant and treats the quadrature purely as a tool for approximating the integral.

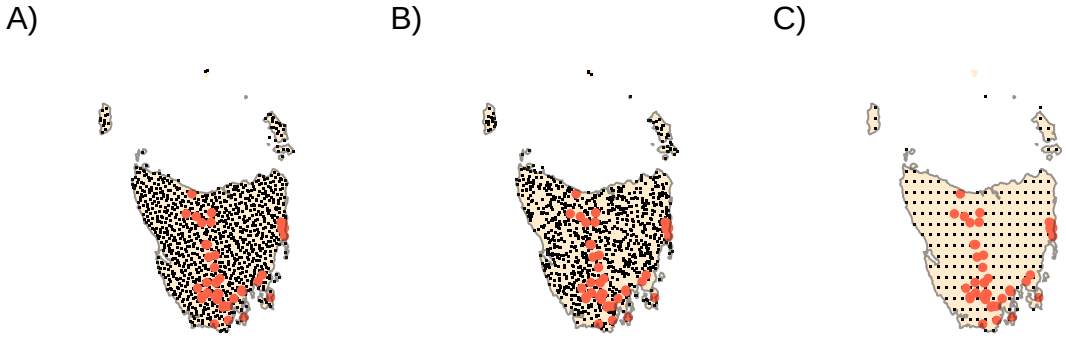

Figure 1: Quadrature schemes generated using the `ppmData` package for the species *Tasmaphena sinclairi* located within Tasmania, Australia. The red points represent the known locations of *Tasmaphena sinclairi*. The black points represent the quadrature locations. A) Quasi-random quadrature where the integration points are generated using a quasi-random areal sample. Weights for each integration point are calculated as the dual of the subsequent Dirichlet tessellation of both the presence and integration points. B) Pseudo-random quadrature generated using random points from within the window. C) Grid quadrature where quadrature points are generated on a regular grid.

Statement of need

There are a number of ways to setup a quadrature scheme to approximate the integral $\Lambda = \int_{s \in \mathcal{A}} \lambda(s) ds$. One common approach is to use a regular grid (Warton & Shepherd, 2010), which creates a regular grid at a known resolution within a spatial window ($\mathcal{A} \subset R^2$). Despite the simplicity of the grid approach, it is not without detraction though. In particular, it is unlikely to handle edge effects well (when there is a consistent and potentially large pattern near the boundaries of $\mathcal{A}$). Then the grid will either be defined away from the edge, missing important trends, or defined on the edge, over-representing this trend. Also, another potential problem stems from the unlikely event that the spatial surface $\lambda(s_i)$ has periodicity in its pattern aligned to the direction of the grid, such as repeated locations of mountain ridges. In that case, the grid-based quadrature will over- or under-estimate the integral depending on whether the grid coincides with peaks or troughs of the surface.

Another common approach is to use a pseudo-random spatial samples and generate approximate areal weights per quadrature point (Phillips et al., 2009; Renner et al., 2015). Typically, this is done by generating pseudo-random points within the study window $\mathcal{A} \subset R^2$ and assuming that all quadrature points have equal weights, generally calculated as $w_i = \frac{m}{|\mathcal{A}|}$ (Renner et al., 2015). This approach is unlikely to suffer either of the potential problems from the grid approach, however it may be inefficient – requiring more points to achieve the same level of accuracy (e.g. Robertson et al., 2013).

Here we use the quasi-random quadrature as a method to trade-off the robustness of pseudo-random sampling and the efficiency of the grid. Quasi-random quadrature spreads the quadrature points in space but simultaneously retains some of the important properties of random sampling. Quasi-random quadrature schemes are available in other R packages, like the excellent `spatstat` (Baddeley, Rubak, & Turner, 2015). The `rQuasi` function can be used for generating a quasi-random spatial sample within a window. However, if the window is not

a rectangle, the number of points will be reduced to include only the ones remaining inside the window boundary, so specifying the number of active points is guess work. Our package `ppmData` allows the exact number of quadrature points to be specified within a spatial window.

The package `spatstat` also allows a user to develop a quasi-random quadrature scheme, but this appears to be quite slow when used with Dirichlet weights for larger numbers of dummy (quadrature) points ($>$ 10000). For example, a regular window with 100000 quadrature points takes approximately 2.5 minutes to run in `spatstat` using `quadscheme` with `method=Dirichlet` with quasi-random dummy points. Our package computes the same Dirichlet tessellation in about 7.5 seconds (Comparisons were from a Linux operating system using a Intel i7 8 core processor, with 16 GB of RAM, see package readme for a quick comparison). Typically, calculating a Dirichlet tessellation increases super-linearly and can become very slow for a large numbers of points. Part of the challenge when developing `ppmData` was to make this step efficient by a C++ implementation of a Delaunay triangulation radial sweep algorithm (Sinclair, 2016). The Dirichlet tessellations are then calculated as the dual graph of the Delaunay triangulation. Areal weights for the quadrature points are calculated as the area of each Dirichlet tessellation polygon.

Generating a quasi-random quadrature scheme

In the `ppmData` package we have tried to make the generation of a quadrature scheme of a single or marked point process as easy as possible. We provide a simple interface for generating a Poisson point process data object and base all inputs and outputs using `terra` (Hijmans, 2021) and `sf` (Pebesma, 2018) packages. Users need to pass a set of coordinates from an observed point pattern, a window to define the point pattern region and a set of covariates observed across this region. The window is defined using a `terra` SpatRaster object, and it is used to identify the point pattern region and where to generate the quadrature locations. Covariates are included as a multiple layered `terra` SpatRaster object that share the same extent and resolution as the window. The values of the covariates will be extracted at the locations of the point pattern and quadrature scheme. Whilst not unique to this package, the `ppmData` package can also generate grid and pseudo-random quadrature schemes as presented in Fig. 1.

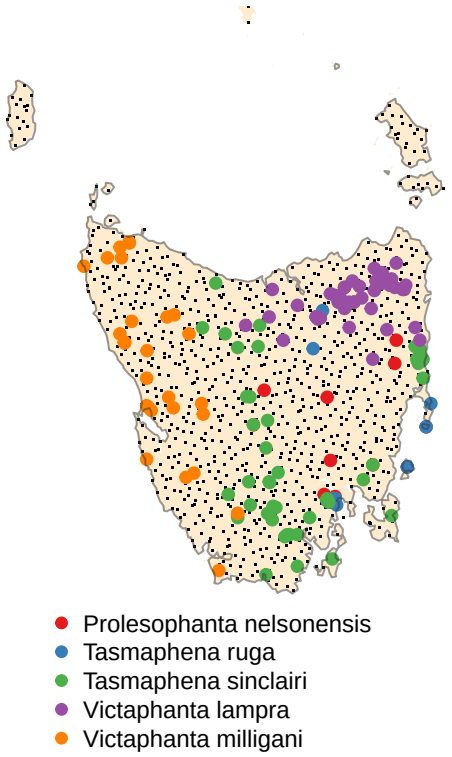

Figure 2: Multiple species quadrature scheme. Here we can see the presences of five different snail species across Tasmania. For the multiple species quadrature there is a common set of background quadrature sites, but the object returns species specific weights.

Multiple species (marked) point process quadrature

One of the main reasons for creating the `ppmData` package was to develop quadrature schemes for multiple species (marked) point process. The multiple species quadrature scheme can be used in multiple species models available for use in the `ecomix` package (Woolley, Foster, & Dunstan, 2022), or models like joint species distribution models (Ovaskainen & Soininen, 2011). Here is a quick example of how we would set up a quasi-random quadrature scheme for multiple species (Fig. 2). The main difference is the column `speciesID` contains multiple species, `ppmData` recognises this internally and subsequently sets up a marked point process quadrature scheme. The quadrature scheme contains a common set of quadrature locations, but the quadrature weights are calculated on a species-specific basis w_{ij}, where i is a site index and j is a species index.

Fitting a model using quadratures generated from `ppmData`

Here we demonstrate how to fit an Inhomogeneous Poisson Process Model using the `glm` function and a `ppmData` object. This approach is very similar to the method presented in Warton & Shepherd (2010), except we are using a quasi-random quadrature. We use `glm` to fit an Inhomogeneous Poisson Process Model for simplicity, but if one wanted to fit an Inhomogeneous Poisson Process Model using different statistical machinery, then one could

– examples are: penalized regressions using `glmnet` (Friedman, Hastie, & Tibshirani, 2010) to achieve a `maxent` analysis (Renner & Warton, 2013), generalized additive models (Wood, 2017) or statistical learning approaches (Hastie, Tibshirani, Friedman, & Friedman, 2009).

```
#load relevant libraries
library(ppmData)
#read the data into R
path <- system.file("extdata", package = "ppmData")
lst <- list.files(path=path,pattern='*.tif',full.names = TRUE)
preds <- rast(lst)
#Format the data (subset and scale)
presences <- subset(snails,SpeciesID %in% "Tasmaphena sinclairi")
preds <- scale(preds)
#Obtain quadrature points, variables and weights
ppmdata <- ppmData(npoints = 20000, presences = presences,
                   window = preds[[1]], covariates = preds)
```

We present a snippet of example code for fitting an Inhomogeneous Poisson Process Model using `glm` from the R `stats` package (R Core Team, 2013). We just need to extract the data.frame from the `ppmData` object and to specify a formula for the model. The only extra step is making the formula response equal to `presence/weights` (which gives us z in the Berman & Turner (1992) approximation). The right side of the formula will represent the covariates and their functional forms. In this example we specify independent 2^{nd} degree polynomials for each covariate, except for the spatial coordinates X & Y which we include in the model as interacting 2^{nd} degree polynomials. Once this is done, we define the weights in the `glm` function call and we now can fit an IPPM using `ppmData` and `glm`.

```
ppp <- ppmdata$ppmData
form  <- presence/weights ~ poly(X,Y, degree = 2) +
                            poly(max_temp_hottest_month, degree = 2) +
                            poly(annual_mean_precip, degree = 2) +
                            poly(annual_mean_temp, degree = 2) +
                            poly(distance_from_main_roads, degree = 2)

ft.ppm <- glm(formula = form, data = ppp,
              weights = as.numeric(ppp$weights),
              family = poisson())
```

Finally, we show how to predict the model using `terra` and a `glm` object. This will return the expected intensity when all raster cell areas equal to one. To get the intensity of *Tasmaphena sinclairi* per cell, we need to re-scale the intensity based on the area represented by each raster cells. The re-scaled intensity returns the expected count of points (species presences) per raster cell. See Fig. 3 for an example output.

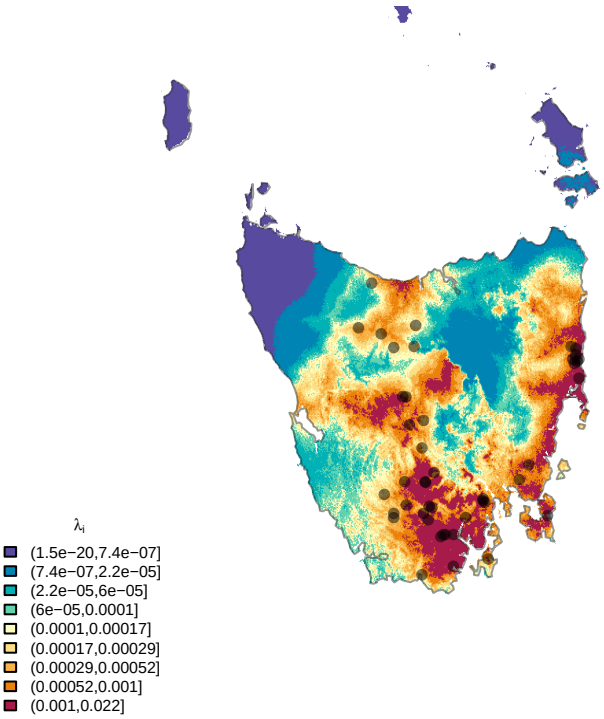

Figure 3: Predicted intensity per for $\approx$ 0.6 km^2 sized raster cell for the snail species *Tasmaphena sinclairi*. The points are occurrence records for *Tasmaphena sinclairi* in the region.

Acknowledgments

We thank Piers Dunstan for comments on an earlier draft of this manuscript.

References

Baddeley, A., Rubak, E., & Turner, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. London: Chapman; Hall/CRC Press. Retrieved from https://www.routledge.com/Spatial-Point-Patterns-Methodology-and-Applications-with-R/Baddeley-Rubak-Turner/9781482210200/

Baddeley, A., & Turner, R. (2000). Practical maximum pseudolikelihood for spatial point patterns. *Australian & New Zealand Journal of Statistics*, *42*(3), 283–322. doi:10.1111/1467-842X.00128

Berman, M., & Turner, T. R. (1992). Approximating point process likelihoods with GLIM. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *41*(1), 31–38. doi:10.2307/2347614

Cressie, N. (1993). *Statistics for spatial data*. John Wiley & Sons.

Diggle, P. J., Menezes, R., & Su, T. (2010). Geostatistical inference under preferential sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *59*(2), 191–232. doi:10.1111/j.1467-9876.2009.00701.x

Foster, S. D., Hosack, G. R., Lawrence, E., Przeslawski, R., Hedge, P., Caley, M. J., Barrett, N. S., et al. (2017). Spatially balanced designs that incorporate legacy sites. *Methods in Ecology and Evolution*, *8*(11), 1433–1442. doi:10.1111/2041-210x.12782

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, *33*(1), 1–22. doi:10.18637/jss.v033.i01

Grafström, A., & Tillé, Y. (2013). Doubly balanced spatial sampling with spreading and restitution of auxiliary totals. *Environmetrics*, *24*(2), 120–131. doi:10.1002/env.2194

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, *2*(1), 84–90. doi:10.1007/bf01386213

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer. doi:10.1007/bf02985802

Hijmans, R. J. (2021). Terra: Spatial Data Analysis. R Package Version 1.1-4.

Liu, J., & Vanhatalo, J. (2020). Bayesian model based spatiotemporal survey designs and partially observed log Gaussian Cox process. *Spatial statistics*, *35*, 100392. doi:10.1016/j.spasta.2019.100392

Ovaskainen, O., & Soininen, J. (2011). Making more out of sparse data: Hierarchical modeling of species communities. *Ecology*, *92*(2), 289–295. doi:10.1890/10-1251.1

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, *10*(1), 439–446. doi:10.32614/RJ-2018-009

Phillips, S. J., Dudík, M., Elith, J., Graham, C. H., Lehmann, A., Leathwick, J., & Ferrier, S. (2009). Sample selection bias and presence-only distribution models: Implications for background and pseudo-absence data. *Ecological Applications*, *19*(1), 181–197. doi:10.1890/07-2153.1

R Core Team. (2013). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org/

Renner, I. W., Elith, J., Baddeley, A., Fithian, W., Hastie, T., Phillips, S. J., Popovic, G., et al. (2015). Point process models for presence-only analysis. *Methods in Ecology and Evolution*, *6*(4), 366–379. doi:10.1111/2041-210X.12352

Renner, I. W., & Warton, D. I. (2013). Equivalence of MAXENT and Poisson point process models for species distribution modeling in ecology. *Biometrics*, *69*(1), 274–281. doi:10.1111/j.1541-0420.2012.01824.x

Robertson, B., Brown, J., McDonald, T., & Jaksons, P. (2013). BAS: Balanced acceptance sampling of natural resources. *Biometrics*, *69*(3), 776–784. doi:10.1111/biom.12059

Sinclair, D. (2016). S-hull: A fast radial sweep-hull routine for Delaunay triangulation. *arXiv preprint arXiv:1604.01428*.

Warton, D., & Shepherd, L. (2010). Poisson point process models solve the "pseudo-absence problem" for presence-only data in ecology. *Annals of Applied Statistics*, *4*(3), 1383–1402. doi:10.1214/10-AOAS331

Wood, S. N. (2017). *Generalized Additive Models: An Introduction with R* (2nd ed.). Chapman; Hall/CRC.

Woolley, S., Foster, S., & Dunstan, P. (2022). Ecomix: Finite mixture models for multiple species grouping of ecological data. Retrieved from https://github.com/skiptoniam/ecomix