INLABru : Convenient digital soil mapping model fitting using INLA-SPDE

Nicolas Saby

Thomas Opitz

4/24/23

Table of contents

1 Introduction

Pedometricians are nowadays big fans of machine learning (ML) approaches with on the top the widely used random forest algorithm, see for example (L. Poggio et al. 2021). These algorithms are indeed particularly adapted to the management of large data sets to map soil properties on wide extent in a large range of situations. The techniques are based on classification and regression, but they take no account of spatial correlations (Heuvelink and Webster 2022). This trend also seems to be accompanied by a lesser use of geostatistical techniques that maybe require more computer resource and statistical skills. However, if prediction is performed in several steps (*eg* regression or any other machine learning prediction in step 1, followed by spatial kriging of the residuals in step 2), then an accurate assessment of the prediction uncertainties is difficult since uncertainties from the first step must be propagated through to the second step.

In this paper, we propose to solve these issues by using the fully Bayesian estimation framework based on the integrated nested Laplace approximation (INLA,(Rue, Martino, and Chopin 2009)), combined with the so-called stochastic partial differential equation approach (SPDE, Lindgren, Rue, and Lindström 2011) providing numerically convenient representations of Gaussian processes over continuous space. The INLA method is an alternative to traditional Markov chain Monte Carlo methods for Bayesian estimation and provides off-the-shelf implementation of fast and accurate deterministic approximations of posterior inferences for a large class of models. INLA with SPDE is handle to handle very large data sets.

INLA-SPDE was already introduced by (Laura Poggio et al. 2016) or (Huang 2017) in the pedometrics community. However, the use of this approach was probably hindered by the complexity of the INLA R package. Recently, the `INLABru` R package (Yuan et al. 2017) originally developed for the point process models has integrates a range of functions to help in implementing INLA-SPDE models in a convenient way. We propose here to show you using a simple and classical regression kriging approach as an example.

When the number of data is huge, it is important to mention that one can improve the performance by using the PARDISO solver. Please, go to https://www.pardiso-project.org/r-inla/#license to apply for a license. Also, use inla.pardiso() for instructions on how to enable the PARDISO sparse library.

2 Set up

2.1 Load packages

We use here a set of R packages in the list below.

The latest version of R (eg >4.2) should be installed on your computer for `INLABru`.

```
library(INLA)
library(inlabru)
library(dplyr)
library(tmap)
library(gstat) # for the meuse data
library(tmap)
library(ggplot2)
```

The `INLABru` method is a wrapper for `INLA::inla` and provides multiple enhancements.

2.2 Point data and rasters

We use the open data `meuse` from the `gstat` package

```
data(meuse)
data(meuse.grid)

str(meuse)
```

```
'data.frame':	155 obs. of  14 variables:
 $ x      : num  181072 181025 181165 181298 181307 ...
 $ y      : num  333611 333558 333537 333484 333330 ...
 $ cadmium: num  11.7 8.6 6.5 2.6 2.8 3 3.2 2.8 2.4 1.6 ...
 $ copper : num  85 81 68 81 48 61 31 29 37 24 ...
 $ lead   : num  299 277 199 116 117 137 132 150 133 80 ...
 $ zinc   : num  1022 1141 640 257 269 ...
 $ elev   : num  7.91 6.98 7.8 7.66 7.48 ...
 $ dist   : num  0.00136 0.01222 0.10303 0.19009 0.27709 ...
 $ om     : num  13.6 14 13 8 8.7 7.8 9.2 9.5 10.6 6.3 ...
 $ ffreq  : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ soil   : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2 1 1 2 ...
 $ lime   : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
 $ landuse: Factor w/ 15 levels "Aa","Ab","Ag",..: 4 4 4 11 4 11 4 2 2 15 ...
 $ dist.m : num  50 30 150 270 380 470 240 120 240 420 ...
```

```r
str(meuse.grid)
```

```
'data.frame':   3103 obs. of  7 variables:
 $ x      : num  181180 181140 181180 181220 181100 ...
 $ y      : num  333740 333700 333700 333700 333660 ...
 $ part.a: num  1 1 1 1 1 1 1 1 1 1 ...
 $ part.b: num  0 0 0 0 0 0 0 0 0 0 ...
 $ dist  : num  0 0 0.0122 0.0435 0 ...
 $ soil  : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ ffreq : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
```

The first action is to create sp objects:

- a SpatialPointsDataFrame corresponding to the regression matrix and,
- the prediction grid

```r
coordinates(meuse) <- c('x','y')

coordinates(meuse.grid) <- c("x","y")
gridded(meuse.grid) = TRUE
```

3 Fully bayesian DSM approach

3.1 The hierarchical model

We construct a hierarchical model in the framework of the SCORPAN approach for the soil property $z(s)$ of a spatial location s. Here z will correspond to the organic matter, om. We will assume the following linkage between model components and observations

$$\eta(s) \sim \text{Intercept} + \underbrace{\sum_{i\in\text{scorpan}} \beta_i^{\text{sc}} z_i^{\text{sc}}(s)}_{\text{SCORPAN factors}} + \underbrace{W(s)}_{\text{Gaussian field}}$$

$\eta(s)$ will then be used in the observation-likelihood,

$$\eta(s)|z(s),\theta \sim \Sigma(\eta(s_i), Q(\theta)^{-1})$$

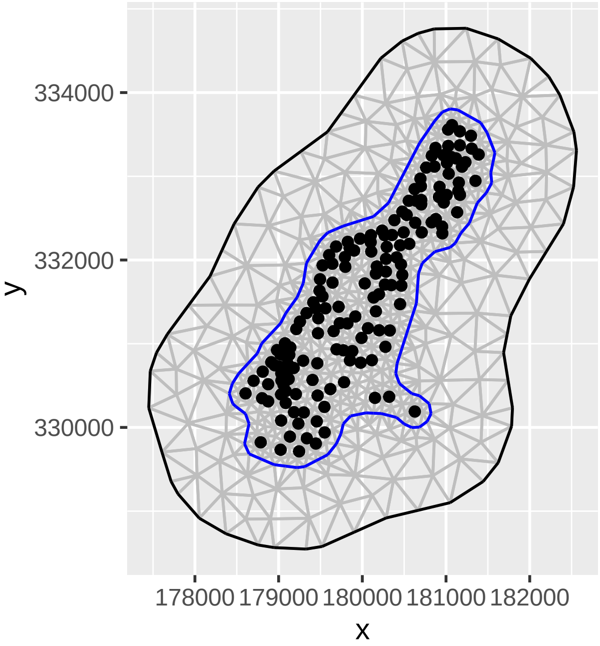
3.2 Construction of the mesh for the SPDE model

INLA and inlabru use a space triangulation method to estimate spatial Gaussian effects with a Matérn covariance function. The spatial Gaussian random field is computed at the mesh nodes by resolving a Stochastic Partial Differential Equation (SPDE), while it is computed elsewhere by interpolation. The mesh definition is based on a trade-off between the finer spatial scale of the spatial effect (higher resolution) and the number of nodes (lower resolution). Below, we present how to build a mesh from the set of coordinates of the calibration sites.

First, we create a matrix `xyMesh` with coordinates of the sites. Next, we define the boundaries of the domain used for computing the spatial latent effect with the SPDE approach. Generally, it is better to compute an internal boundary and an external boundary with different resolutions to address boundary effects.

The `INLA::inla.mesh.2d` function creates a triangle mesh based on initial point locations, specified or automatic boundaries, and mesh quality parameters, in particular the `cutoff`. Some information here: https://rpubs.com/jafet089/886687

```
cutoffValue = 50 # in meter

xyMesh <- rbind(coordinates(meuse)) # transform into matrix

max.edge = diff(range(xyMesh[,1]))/(3*5)
bound.outer = diff(range(range(xyMesh[,1])))/3

bndint <- inla.nonconvex.hull(meuse, convex=-.05)
bndext <- inla.nonconvex.hull(meuse, convex=-.3)

# Use of inla.mesh.2d
mesh = inla.mesh.2d(loc=xyMesh,
                    boundary = list(int = bndint,
                                    out = bndext),
                    max.edge = c(1,3)*max.edge,
                    cutoff = cutoffValue,
                    crs = meuse@proj4string@projargs)
ggplot() +
  gg(mesh) +
  gg(meuse) +
  coord_equal()
```



3.3 Defining the spatial Gaussian random field $W(s)$

We choose the Matérn covariance function for the Gaussian random field because it can be easily fitted in `INLA` using a SPDE. The Matérn covariance in `INLA` depends on three parameters: - a fractional order parameter *alpha* in the SPDE linked to the smoothness of the solution, - a standard deviation parameter *sigma* and, - a spatial correlation parameter known as the *range*.

We specify these parameters in our model by selecting a penalized complexity prior using the `INLA::inla.spde2.pcmatern` function. For more details, please refer to the introduction to spatial models with `INLA` in chapter 7 at <https://becarioprecario.bitbucket.io/inla-gitbook/ch-spatial.html>.

```
matern <-
  INLA::inla.spde2.pcmatern(mesh,
                            alpha = 2,# fractional operator which is related
                            prior.sigma = c(1, 0.5),# P(sigma > 1) = 0.5
                            prior.range = c(10000, 0.9)  # P(range < 10000 m) = 0.9
  )
```

3.4 Specify the hierarchical model

We then specify, in `cmp`, the model components using the convinient `INLA Bru` approach. We use as example the following latent effects: intercept, a linear relationship with the covariate corresponding to the distance to the river, and the Gaussian random field.

```
cmp <- om ~
  field(coordinates, model = matern ) +
  Intercept(1) +
  dist(dist, model = 'linear' )
```

Finally, we fit the hierarchical model to the data using the `bru` function of the `inlabru` package. This function requires the model components defined earlier (`cmp`), the dataset (`meuse`), , the mesh (`mesh`) where the model will be evaluated, and several options to control the INLA algorithm.

the spatial domain where the data were collected can be aslo provided using the (`domainSP`)

We use here the `eb` strategy as it is much quicker to compute but a bit less accurate.

```
fit <- inlabru:: bru(components = cmp,
           data = meuse,
           family = "gaussian",
           domain = list(coordinates = mesh),
           options = list(
             control.inla = list(int.strategy = "eb"),
             verbose = FALSE)
           )
```

The summary gives the posterior estimates of fixed effects (intercept and elevation) and hyperparameters (standard deviation and range of the Gaussian random field).

We can look at some summaries of the posterior distributions for the parameters, for example the fixed effects (i.e. the intercept) and the hyper-parameters (i.e. dispersion in the gamma likelihood, the precision of the RW1, and the parameters of the spatial field):

```
summary(fit)
```

```
inlabru version: 2.7.0
```

```
INLA version: 22.12.16
Components:
field: main = spde(coordinates)
Intercept: main = linear(1)
dist: main = linear(dist)
Likelihoods:
  Family: 'gaussian'
    Data class: 'SpatialPointsDataFrame'
    Predictor: om ~ .
Time used:
    Pre = 1.3, Running = 0.847, Post = 0.0666, Total = 2.21
Fixed effects:
             mean    sd 0.025quant 0.5quant 0.975quant    mode kld
Intercept  10.976 1.149      8.723   10.976     13.228  10.976   0
dist      -11.587 2.607    -16.696  -11.587     -6.478 -11.587   0

Random effects:
  Name    Model
    field SPDE2 model

Model hyperparameters:
                                            mean      sd 0.025quant 0.5quant
Precision for the Gaussian observations    0.338   0.084      0.203    0.328
Range for field                         1070.631 423.165    522.243  979.530
Stdev for field                            3.191   0.755      2.040    3.078
                                        0.975quant    mode
Precision for the Gaussian observations      0.532   0.309
Range for field                           2150.174 820.388
Stdev for field                              4.991   2.837

Deviance Information Criterion (DIC) ...............: 669.11
Deviance Information Criterion (DIC, saturated) ....: 217.11
Effective number of parameters .....................: 61.64

Watanabe-Akaike information criterion (WAIC) ...: 670.85
Effective number of parameters .................: 49.87

Marginal log-Likelihood:  -380.49 
 is computed 
Posterior summaries for the linear predictor and the fitted values are computed
(Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

4 Spatial predictions

Now we use the fit to predict the field on a lattice, and generate a set of results using 100 realizations from the posterior distribution:

```
pred <- predict(
  fit,
  n.samples = 100,
  meuse.grid,
  ~ field + Intercept + dist ,
  num.threads = 2
)
```

It is also very simple to draw samples from the posterior distribution. Here we draw 5 samples and select the first one.

```
samp <- generate(fit,
                 meuse.grid,
                 ~ field + Intercept + dist ,
                 n.samples = 5
)

str(samp)
```

```
 num [1:3103, 1:5] 12.5 13.5 12.4 11 14.3 ...
```

```
pred$sample <- samp[, 1]
```

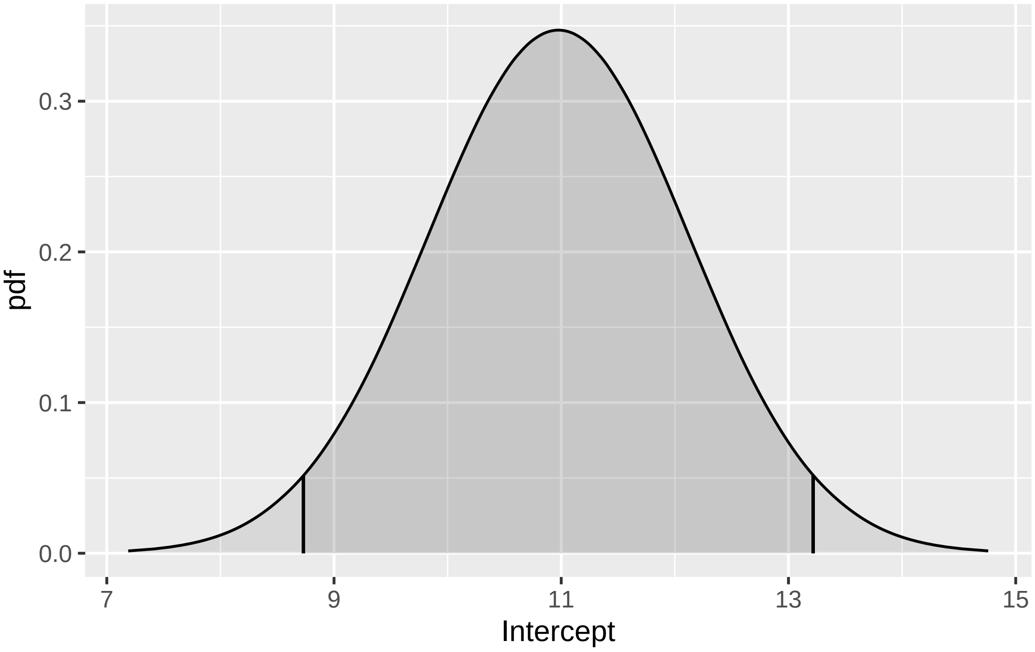
5 Plotting results

5.1 The differents effects

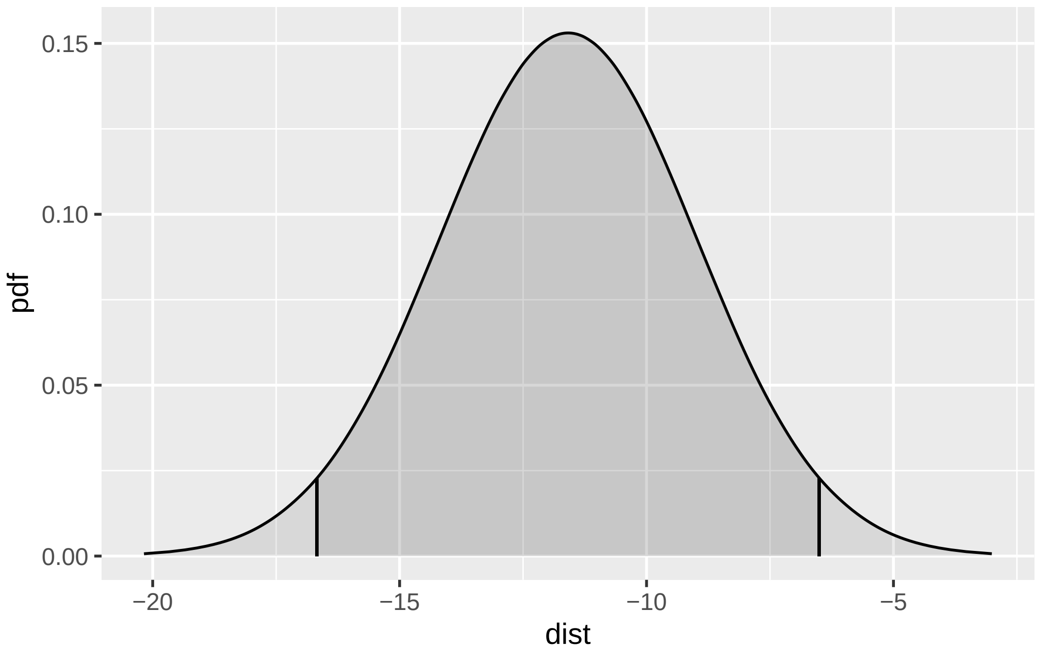
We can plot the posterior densities for the latent effect Intercept and distance to the border.

To this end we will use the `inlabru::plot()` function,

```
plot(fit, "Intercept")
```



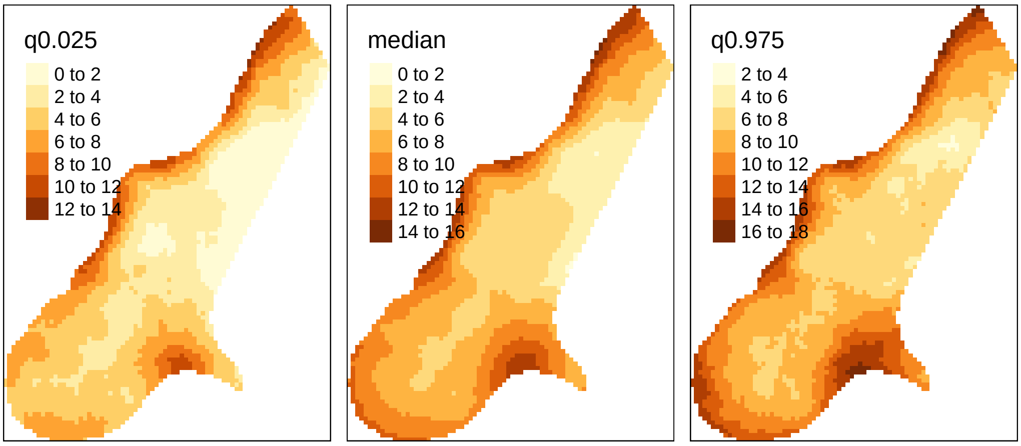
```
plot(fit, "dist")
```



5.2 The spatial prediction maps with uncertainty

You can plot the median, lower 95% and upper 95% density surfaces as follows (assuming that the predicted intensity is in object `pred`).

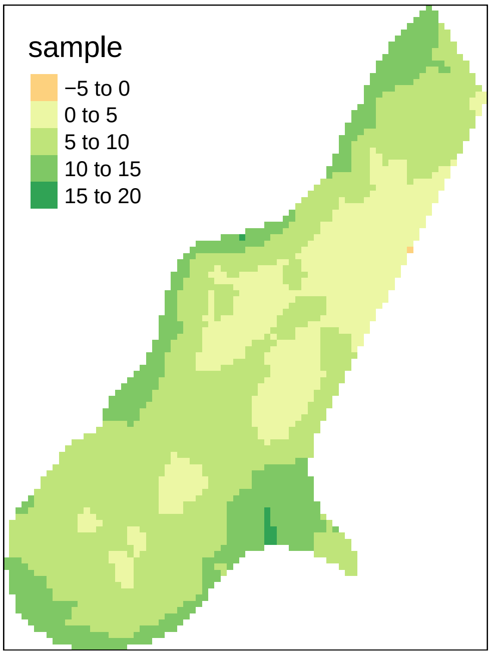
```
pred$q0.025[pred$q0.025<0] = 0

tm_shape(pred) +
  tm_raster(
    c("q0.025","median","q0.975")
    )
```



5.3 One realisation

The sample from the posterior distribution

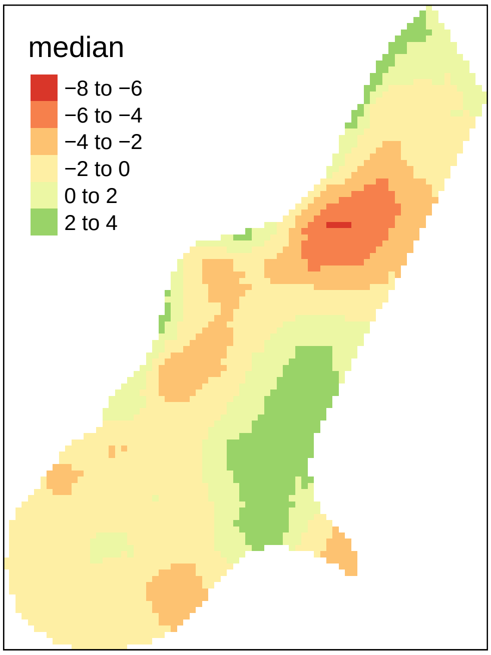
```
tm_shape(pred) + tm_raster("sample")
```



6 The spatial effect

We plot here the spatial Gaussian random field $W(s)$

```
pred <- predict(
  fit,
  n.samples = 100,
  meuse.grid,
  ~ field  ,
  num.threads = 2
)

tm_shape(pred) + tm_raster("median")
```



7 rSPDE fit

add this here ?

```
library(rSPDE)
rspde_model <- rspde.matern(mesh = mesh)
cmp <- om ~
  field(coordinates, model = rspde_model ) +
  Intercept(1) +
  dist(dist, model = 'linear' )
fit2 <- inlabru:: bru(components = cmp,
           data = meuse,
           family = "gaussian",
           domain = list(coordinates = mesh),
           options = list(
             control.inla = list(int.strategy = "eb"),
             verbose = FALSE)
           )
result_fit <- rspde.result(fit2, "field", rspde_model)
summary(result_fit)
```

```
posterior_df_fit <- gg_df(result_fit)

ggplot(posterior_df_fit) + geom_line(aes(x = x, y = y)) +
facet_wrap(~parameter, scales = "free") + labs(y = "Density")

pred <- predict(
  fit,
  meuse.grid,
  ~ field + Intercept + dist ,
  num.threads = 2
)
pred$q0.025[pred$q0.025<0] = 0
tm_shape(pred) +
  tm_raster(
    c("q0.025","median","q0.975")
    )
```

8 Code availability

The code is also available on github : https://github.com/nsaby/pedometron042023

9 References

Heuvelink, Gerard B. M., and Richard Webster. 2022. "Spatial Statistics and Soil Mapping: A Blossoming Partnership Under Pressure." *Spatial Statistics* 50: 100639. https://doi.org/https://doi.org/10.1016/j.spasta.2022.100639.

Huang, Malone, J. 2017. "Evaluating a Bayesian modelling approach (INLA-SPDE) for environmental mapping." *Science of The Total Environment* 609: 621--632.

Lindgren, Finn, Håvard Rue, and Johan Lindström. 2011. "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73 (4): 423–98.

Poggio, Laura, Alessandro Gimona, Luigi Spezia, and Mark J Brewer. 2016. "Bayesian spatial modelling of soil properties and their uncertainty: The example of soil organic matter in Scotland using R-INLA." *Geoderma* 277: 69–82.

Poggio, L., L. M. de Sousa, N. H. Batjes, G. B. M. Heuvelink, B. Kempen, E. Ribeiro, and D. Rossiter. 2021. "SoilGrids 2.0: Producing Soil Information for the Globe with Quantified Spatial Uncertainty." *SOIL* 7 (1): 217–40. https://doi.org/10.5194/soil-7-217-2021.

Rue, Håvard, Sara Martino, and Nicolas Chopin. 2009. "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations." *Journal of the Royal Statistical Society: Series b (Statistical Methodology)* 71 (2): 319–92.

Yuan, Y., F. E. Bachl, F. Lindgren, D. L. Brochers, J. B. Illian, S. T. Buckland, H. Rue, and T. Gerrodette. 2017. "Point Process Models for Spatio-Temporal Distance Sampling Data from a Large-Scale Survey of Blue Whales." https://arxiv.org/abs/1604.06013.