MRSea: 2D Interaction Example

LAS Scott-Hayward, CG Walker and ML Mackenzie

2021-07-19

This vignette is in addition to the main vignette for the package and shows an analysis which allows different knot locations for different levels of a spatial interaction term. This may be particularly useful where the spatial coverage of the two levels is not the same.

Load data

1. Load the distance sampling corrected Nysted data from the package.

```
data("nysted.analysisdata")
mydata<-nysted.analysisdata # renamed just to be less typing!
```

2. Fit an initial Model to the data. Here we include only the `impact` variable as a factor along with the offset of cell area.

```
mydata$blockid <- paste(mydata$transect.id, mydata$season, mydata$impact,sep = "")
```

Make distance matrices

3. Create a grid of knots that will be used as possible knot locations. Create one grid for before and one for after. As we have complete spatial overlap between the two `impact` levels, I have chosen the same number of candidate knot locations for each.

```
nk=150 # number of knots per interaction level
myknots<-NULL
for(i in 1:length(unique(mydata$impact))){
  faclevel<-unique(mydata$impact)[i]
  tempkg<-getKnotgrid(coordData = cbind(mydata$x.pos[mydata$impact==faclevel],
                                        mydata$y.pos[mydata$impact==faclevel]),
                  numKnots = nk, plot = FALSE)
  myknots<-rbind(myknots, data.frame(tempkg, impact =paste(faclevel)))
}
```

```
head(myknots)
#>        X1      X2 impact
#> 1 678.25 6035.75      0
#> 2 688.25 6034.75      0
#> 3 659.75 6054.25      0
#> 4 674.25 6057.25      0
#> 5 700.25 6044.75      0
#> 6 696.25 6043.75      0
```

```
require(ggplot2)
ggplot() +
  geom_point(data=mydata, aes(x.pos, y.pos)) +
  geom_point(data=myknots, aes(X1, X2), colour='red') +
  facet_wrap(~impact, nrow=2) +
  theme_bw() + xlab('Easting (Km)') + ylab('Northing (Km)') +
    coord_equal()
```
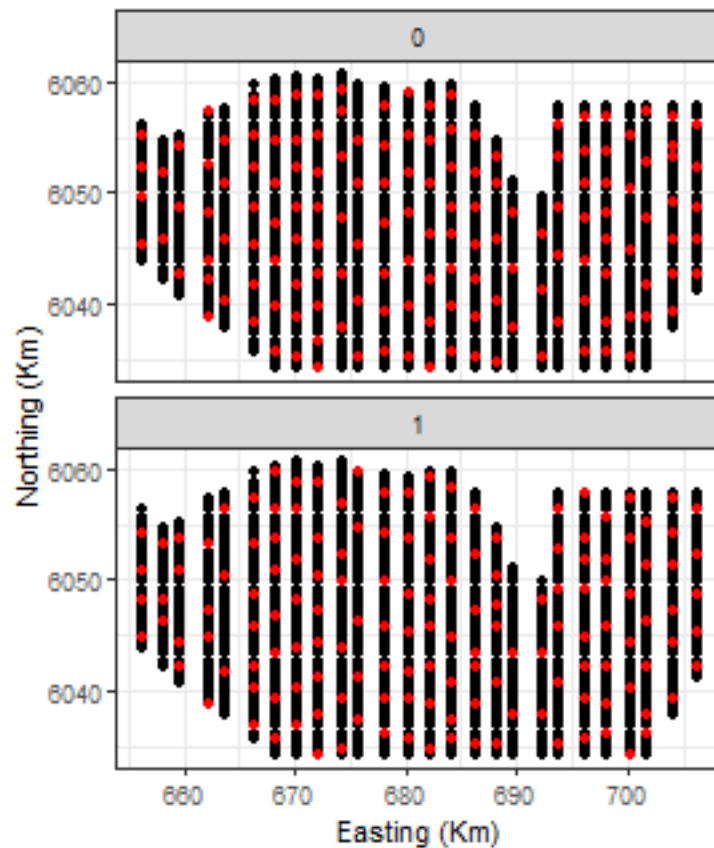

Figure 1: Figure showing the candidate knot locations for each of the interaction levels

The `myknots` object has three columns (x, y and impact) but it is also useful to have an object with just the coordinates.

```
kg<-myknots[,1:2]
```

4. Create distances between knots and data. At some future point, this ability will be a function in the package.

```
require(dplyr)
dists<-makeDists(datacoords = mydata[,c('x.pos', 'y.pos')], knotcoords = kg, knotmat = TRUE)
d2k = dists$dataDist
k2k = dists$knotDist

n.level<-length(unique(myknots[,3]))
```

```r
nk.in.level<-data.frame(myknots %>%
                        mutate(id = row_number()) %>%
                        group_by(impact) %>%
                        summarise(beg=first(id), end=last(id)))
k2k_temp<-matrix(Inf, nrow = nrow(k2k), ncol = ncol(k2k))
for(i in 1:n.level){
  kid<-nk.in.level[i,'beg']:nk.in.level[i,'end']
  k2k_temp[kid,kid]<-k2k[kid, kid]
}
k2k<-k2k_temp

n.level<-length(unique(myknots[,3]))
nk.in.level<-data.frame(myknots %>% mutate(id = row_number()) %>%
                        group_by(impact) %>%
                        summarise(beg=first(id), end=last(id)))
ncoords.in.level <- data.frame(mydata %>%
                               mutate(id = row_number()) %>%
                               group_by(impact) %>% summarise(beg=first(id), end=last(id)))
d2k_temp<-matrix(Inf, nrow = nrow(d2k), ncol = ncol(d2k))
for(i in 1:n.level){
  kid<-nk.in.level[i,'beg']:nk.in.level[i,'end']
  did<-ncoords.in.level[i,'beg']:ncoords.in.level[i,'end']
  d2k_temp[did,kid]<-d2k[did, kid]
}
d2k<-d2k_temp
```

Analysis

5. Run SALSA2D to find the appropriate number and location of knots for the 2D smooth term of `x.pos` and `y.pos`. For speed of fitting, I am specifying a small number of knots and a quick fitness measure, QAIC. Remember that as we are using the distance matrix to act as the interaction, we do not specify the interaction in the `salsa2dlist` object but we do include `impact` as a main effect in the initial model. Additionally, the knotgrid specified in `salsa2dlist` must have three columns (x, y, interaction level).

```r
initialModel <- glm(response ~ as.factor(season) + as.factor(impact)
                    + offset(log(area)), family = "quasipoisson",
                    data = mydata)
```

```r
# make parameter set for running salsa2d
salsa2dlist<-list(fitnessMeasure = "QAIC",
                  knotgrid = myknots,
                  startKnots=10,
                  minKnots=4,
                  maxKnots=12)
```

```r
salsa2dOutput<-runSALSA2D(model = initialModel,
                          salsa2dlist = salsa2dlist,
                          d2k=d2k,k2k=k2k)
```

6. Assess the output for the location of the selected knots.

```
mymodel<-salsa2dOutput$bestModel
chosenknots <- myknots[mymodel$splineParams[[1]]$knotPos,]
```

```
count(chosenknots, impact)
#>   impact n
#> 1      0 4
#> 2      1 7
#startingknots <- myknots[startknotlocs,]
```

```
imp.labs <- c("Pre-Construction", "Post-Construction")
names(imp.labs) <- c("0", "1")

# quick look to see what was chosen
ggplot(myknots) +
  geom_point(aes(x=X1, y=X2)) +
  geom_point(aes(x=X1, y=X2, size=2), data=chosenknots, alpha=4/5,
             show.legend = FALSE, shape=5) +
  theme_bw() + xlab('Easting (Km)') + ylab('Northing (Km)') +
  coord_equal() +
  facet_wrap(~impact, ncol=1, labeller = labeller(impact=imp.labs))
```
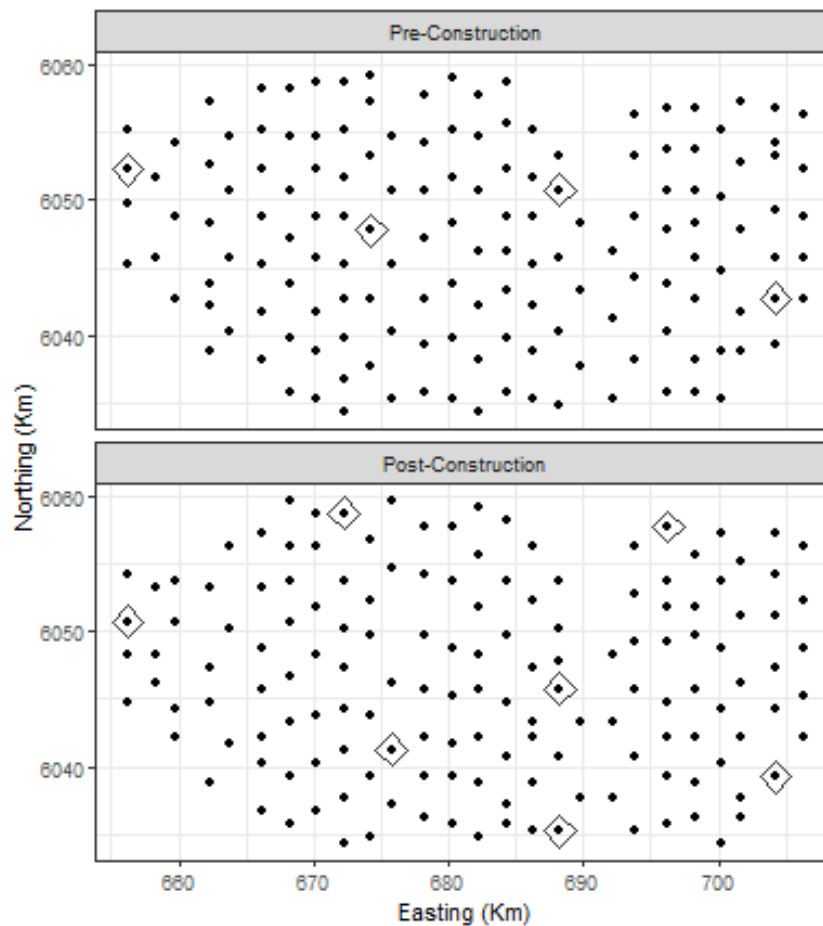

7. You can find the knot locations chosen by querying model object.

```
mymodel$splineParams[[1]]$knotPos
#>  [1]   87 277 211 237   70   12 157 227 207 230   42
```

```
set.seed(1)
cv.gamMRSea(mydata, mymodel, K=10)$delta[1]
#> [1] 5.968254
```

Making predictions

8. To make predictions to a grid, we also need to create a prediction distance matrix and we do this in the same way as for the data above.

```
data("nysted.predictdata")
datacoords<-nysted.predictdata[,c('x.pos', 'y.pos', 'impact')]
dists<-makeDists(datacoords = datacoords[,1:2], knotcoords = kg, knotmat = FALSE)
g2k = dists$dataDist

n.level<-length(unique(myknots[,3]))
nk.in.level<-data.frame(myknots %>%
                        mutate(id = row_number()) %>%
                        group_by(impact) %>%
                        summarise(beg=first(id), end=last(id)))
ncoords.in.level <- data.frame(datacoords %>%
                        mutate(id = row_number()) %>%
                        group_by(impact) %>%
                        summarise(beg=first(id), end=last(id)))
g2k_temp<-matrix(Inf, nrow = nrow(g2k), ncol = ncol(g2k))
for(i in 1:n.level){
  kid<-nk.in.level[i,'beg']:nk.in.level[i,'end']
  gid<-ncoords.in.level[i,'beg']:ncoords.in.level[i,'end']
  g2k_temp[gid,kid]<-g2k[gid, kid]
}
g2k<-g2k_temp
```

```
# make predictions on response scale
nysted.predictdata$preds<-predict.gamMRSea(newdata = nysted.predictdata,
                                           g2k =g2k,
                                           object = mymodel)
```
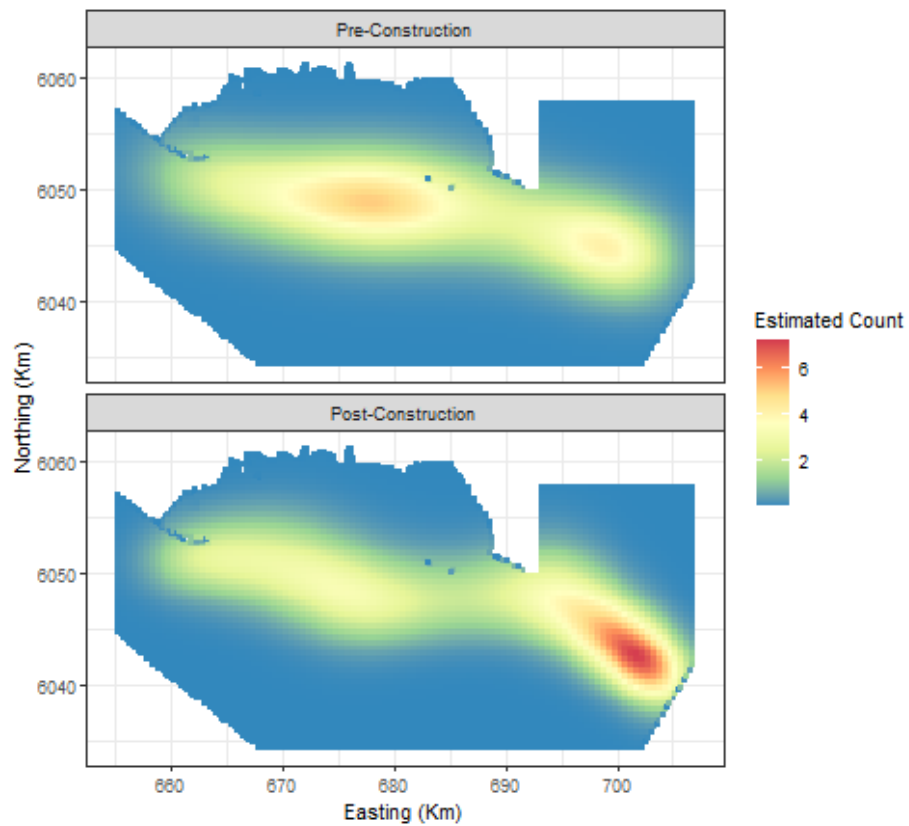
Plotting the predictions pre and post impact:

```
ggplot() +
  geom_tile(aes(x=x.pos, y=y.pos, fill=preds), height=0.5, width=0.5,
            data=filter(nysted.predictdata, season==1)) +
  coord_equal()+
  scale_fill_distiller(palette = "Spectral",name="Estimated Count") +
  theme_bw() +
  xlab('Easting (Km)') + ylab('Northing (Km)') +
  #geom_polygon(data=nysted.coast, aes(x, y), fill='grey') +
  facet_wrap(~impact, ncol=1, labeller = labeller(impact = imp.labs))
```

and, since this is simulated data, we can also look at the truth:

```
ggplot() +
  geom_tile(aes(x=x.pos, y=y.pos, fill=truth.re), height=0.5, width=0.5,
            data=filter(nysted.predictdata, season==1)) +
  coord_equal()+
  scale_fill_distiller(palette = "Spectral",name="Estimated Count") +
  theme_bw() +
  xlab('Easting (Km)') + ylab('Northing (Km)') +
  #geom_polygon(data=nysted.coast, aes(x, y), fill='grey') +
  facet_wrap(~impact, ncol=1, labeller = labeller(impact = imp.labs))
```
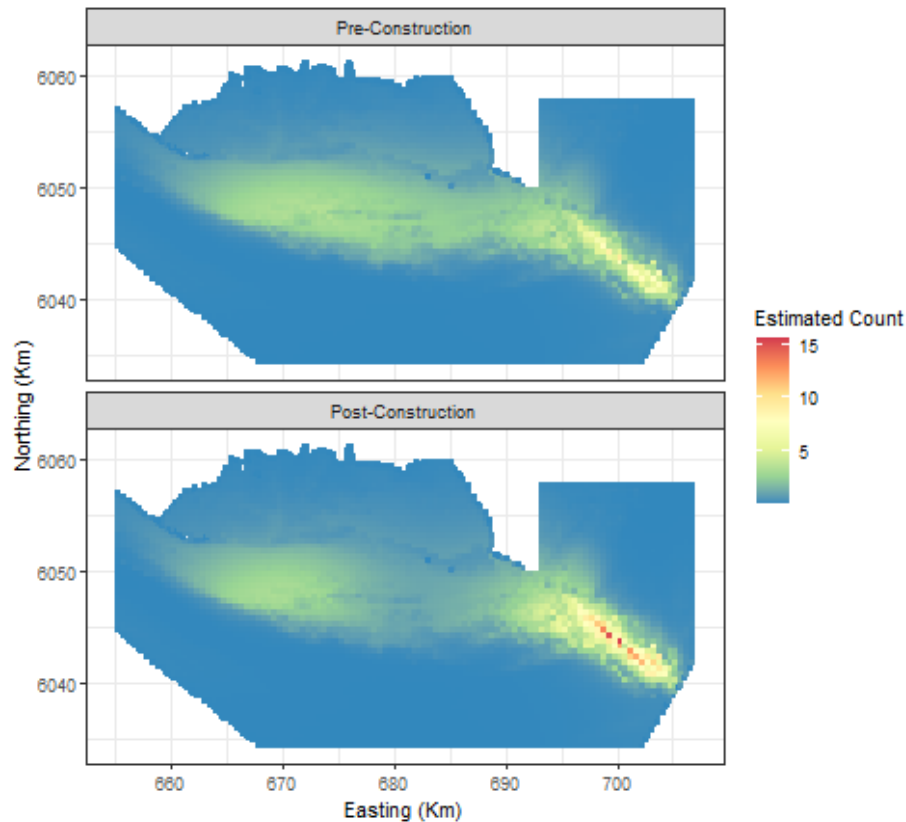
Choosing your own starting knot locations

In the above analysis, the SALSA algorithm will space-fill across all knot locations to initialise the first set of knots. This could mean that in the initial step, one of the levels gets very few starting locations. If you wish to have a more even set of starting locations, you can specify these locations. Here I am specifying 5 locations from each level of the interaction.

```
startknotlocs<-c(
    as.numeric(rownames(getKnotgrid(filter(myknots, impact == 0)[,1:2],
                                    numKnots = 5, plot = FALSE))),
    as.numeric(rownames(getKnotgrid(filter(myknots, impact == 1)[,1:2],
                                    numKnots = 5, plot = FALSE))) + 150)
```

The set up of the model is a little different and I have included **##** below to indicate which lines have been changed/added.

```
salsa2dlist <- list(fitnessMeasure = 'QAIC',
                    knotgrid = myknots,
                    startKnots = length(startknotlocs), ##
                    minKnots = 4,
                    maxKnots = 12,
                    gap = 0)
```

```
salsa2doutput_stkn<-runSALSA2D(model = initialModel,
                               salsa2dlist = salsa2dlist,
                               d2k= d2k, k2k = k2k,
                               initialise=FALSE,  ##
                               initialKnPos = startknotlocs, ##
                               suppress.printout = TRUE)
```

As before, we can look to see where the model has placed the knots. The starting locations are a grey box with a cross and the final locations are red diamonds

```
mymodel_sk<-salsa2doutput_stkn$bestModel
chosenknots_sk <- myknots[mymodel_sk$splineParams[[1]]$knotPos,]
startingknots <- myknots[startknotlocs,]
```

```
count(chosenknots_sk, impact)
```

```
# quick look to see what was chosen
ggplot(myknots) +
  geom_point(aes(x=X1, y=X2)) +
  geom_point(aes(x=X1, y=X2, size=2),data=chosenknots_sk, alpha=4/5,
             show.legend = FALSE, shape=5, colour = 'firebrick') +
  geom_point(aes(x=X1, y=X2, size=2), shape = 7, data=startingknots,
             colour = 'darkgrey', show.legend = FALSE) +
  theme_bw() + xlab('Easting (Km)') + ylab('Northing (Km)') +
  coord_equal() +
  facet_wrap(~impact, ncol=1, labeller = labeller(impact=imp.labs))
```

```
set.seed(1)
cv.gamMRSea(mydata, mymodel_sk, K=10)$delta[1]
#> [1] 5.932845
```

This model is better than the earlier one which used a space-filled selection of all knots for starting locations. However, bear in mind that both models are using a simplistic number of start knots and could be improved.

```
# make predictions on response scale
nysted.predictdata$preds_sk<-predict.gamMRSea(newdata = nysted.predictdata,
                                               g2k =g2k,
                                               object = mymodel_sk)
```

Plotting the predictions pre and post impact:

```
ggplot() +
  geom_tile(aes(x=x.pos, y=y.pos, fill=preds_sk), height=0.5, width=0.5,
            data=filter(nysted.predictdata, season==1)) +
  coord_equal()+
  scale_fill_distiller(palette = "Spectral",name="Estimated Count") +
  theme_bw() +
  xlab('Easting (Km)') + ylab('Northing (Km)') +
  #geom_polygon(data=nysted.coast, aes(x, y), fill='grey') +
  facet_wrap(~impact, ncol=1, labeller = labeller(impact = imp.labs))
```
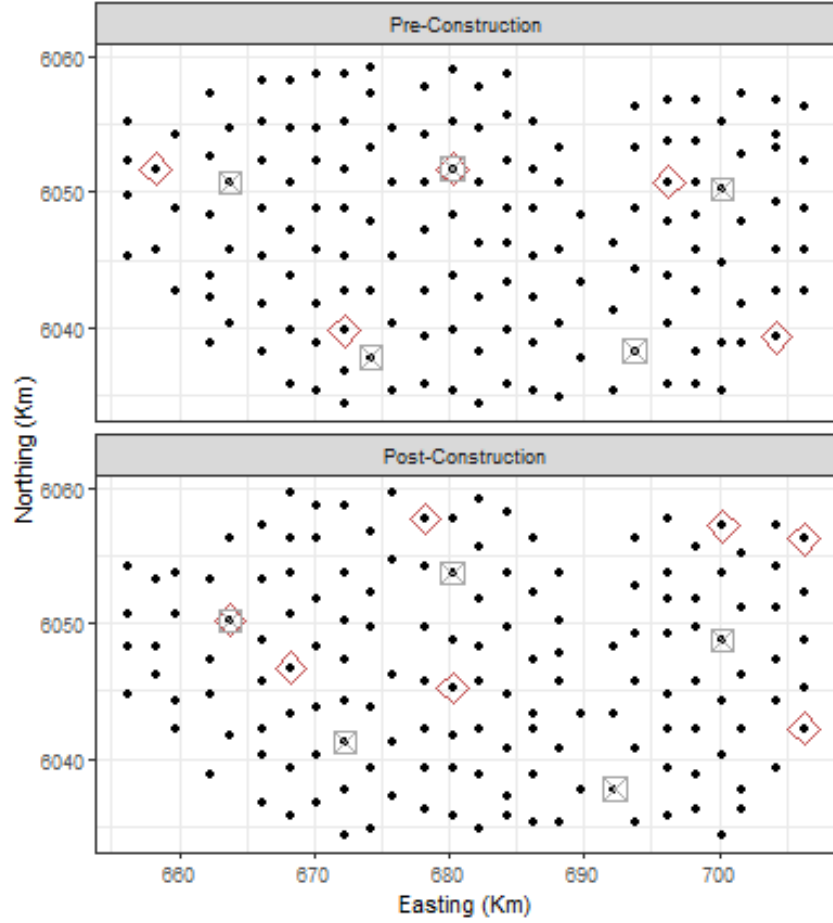
Figure 2: Black dots are candidate knot locations, grey squares with crosses are initial knot locations and red diamonds are final knot locations.