

Advanced topics in distance sampling

Workshop, 26-30 August 2019

Centre for Research into Ecological and Environmental Modelling

Exercise 6. Prediction using fitted density surface models

Now we've fitted some models, let's use the `predict` functions and the data from GIS to make predictions of abundance.

Aims

By the end of this practical, you should feel comfortable:

- Loading raster data into R
- Building a `data.frame` of prediction covariates
- Making a prediction using the `predict()` function
- Summing the prediction cells to obtain a total abundance for a given area
- Plotting a map of predictions
- Saving predictions to a raster to be used in ArcGIS

Loading the packages and data

```
library(knitr)
library(dsm)

## Loading required package: mgcv
## Loading required package: nlme
## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.
## Loading required package: mrds
## This is mrds 2.2.1
## Built: R 3.6.1; ; 2019-07-17 13:15:38 UTC; windows
## Loading required package: numDeriv
## This is dsm 2.2.17
## Built: R 3.6.1; ; 2019-07-11 19:38:05 UTC; windows

library(ggplot2)
# colourblind-friendly colourschemes
library(viridis)

## Loading required package: viridisLite
```

```

# to load and save raster data
library(raster)

## Loading required package: sp

##
## Attaching package: 'raster'

## The following object is masked from 'package:nlme':
##
##      getData

# models with only spatial terms
load("dsms-xy.RData")
# models with all covariates
load("dsms.RData")

```

Loading prediction data

Before we can make predictions, we first need to load the covariates into a “stack” from their files on disk using the `stack()` function from `raster`. We give `stack()` a vector of locations to load the rasters from. Note that in RStudio you can use tab-completion for these locations and avoid some typing. At this point we arbitrarily choose the time periods of the SST, NPP and EKE rasters (2 June 2004, or Julian date 153).

```

predictorStack <- stack(c("Covariates_for_Study_Area/Depth.img",
                          "Covariates_for_Study_Area/GLOB/CMC/CMC0.2deg/analysed_sst/",
                          "Covariates_for_Study_Area/VGPM/Rasters/vgpm.2004153.hdf.gz",
                          "Covariates_for_Study_Area/DistToCanyonsAndSeamounts.img",
                          "Covariates_for_Study_Area/Global/DT\\ all\\ sat/MSLA_ke/2004
                          ))

```

We need to rename the layers in our stack to match those in the model we are going to use to predict. If you need a refresher on the names that were used there, call `summary()` on the DSM object.

```

names(predictorStack) <- c("Depth", "SST", "NPP", "DistToCAS", "EKE")

```

Now these are loaded, we can coerce the stack into something `dsm` can talk to using the `as.data.frame` function. Note we need the `xy=TRUE` to ensure that `x` and `y` are included in the prediction data. We also set the offset value, i.e. the area of each cell in our prediction grid.

```

predgrid <- as.data.frame(predictorStack, xy=TRUE)
predgrid$off.set <- (10*1000)^2

```

We can then predict for the model `dsm_nb_xy_ms`:

```

pp <- predict(dsm_nb_xy_ms, predgrid)

```

This is just a list of numbers – the predicted abundance per cell. We can sum these to get

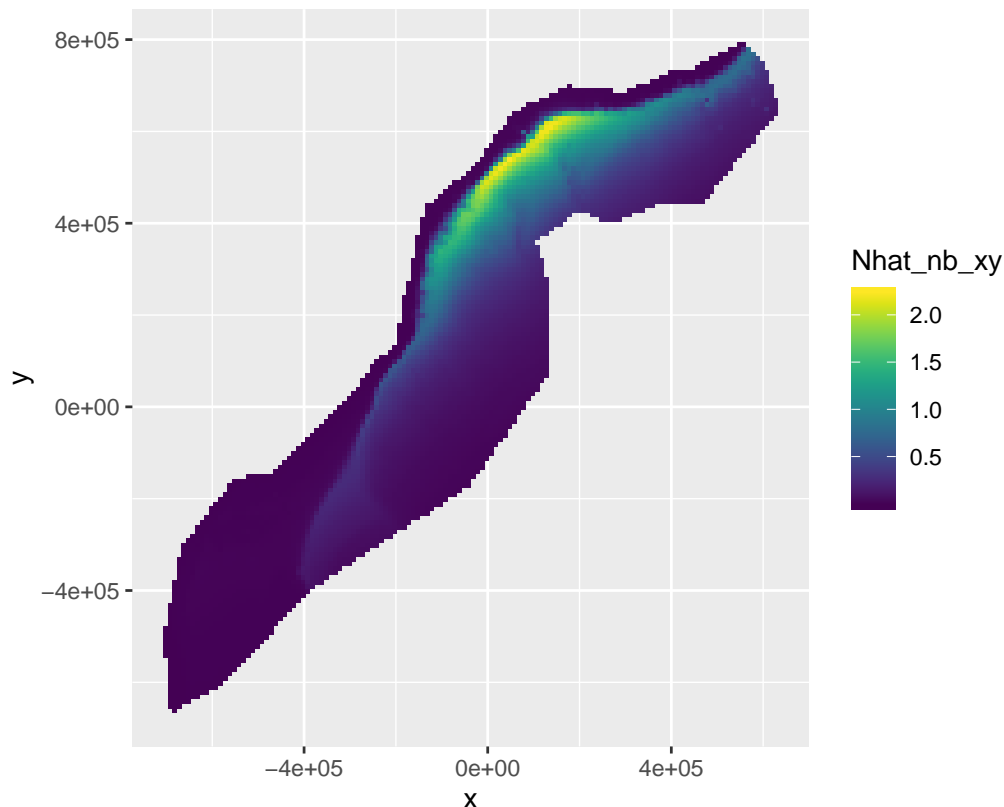


Figure 1: Predicted surface for abundance estimates with bivariate spatial smooth along with environmental covariates.

the estimated abundance for the study area:

```
sum(pp, na.rm=TRUE)
```

```
## [1] 1589.216
```

Because we predicted over the whole raster grid (including those cells without covariate values – e.g. land), some of the values in `pp` will be NA, so we can ignore them when we sum by setting `na.rm=TRUE`. We need to do this again when we plot the data too.

We can also plot this to get a spatial representation of the predictions:

```
# assign the predictions to the prediction grid data.frame
predgrid$Nhat_nb_xy <- pp
# remove the NA entries (because of the grid structure of the raster)
predgrid_plot <- predgrid[!is.na(predgrid$Nhat_nb_xy),]
# plot!
p <- ggplot(predgrid_plot) +
  geom_tile(aes(x=x, y=y, fill=Nhat_nb_xy, width=10*1000, height=10*1000)) +
  coord_equal() +
  scale_fill_viridis()
print(p)
```

Copy the chunk above and make predictions for the other models you saved in the previous exercises. In particular, compare the models with only spatial terms to those

with environmental covariates included.

Save the prediction to a raster

To be able to load our predictions into ArcGIS, we need to save them as a raster file. First we need to make our predictions into a raster object and save them to the stack we already have:

```
# setup the storage for the predictions
pp_raster <- raster(predictorStack)
# put the values in, making sure they are numeric first
pp_raster <- setValues(pp_raster, as.numeric(pp))
# name the new, last, layer in the stack
names(pp_raster) <- "Nhat_nb_xy"
```

We can then save that object to disk as a raster file:

```
writeRaster(pp_raster, "abundance_raster.img", datatype="FLT4S", overwrite=TRUE)
```

Here we just saved one raster layer: the predictions from model `Nhat_nb_xy`. Try saving another set of predictions from another model by copying the above chunk.

You can check that the raster was written correctly by using the `stack()` function, as we did before to load the data and then the `plot()` function to see what was saved in the raster file.

Save prediction grid to RData

We'll need to use the prediction grid and predictor stack again when we calculate uncertainty in the next practical, so let's save those objects now to save time later.

```
save(predgrid, predictorStack, file="predgrid.RData")
```

Extra credit

- Try refitting your models with `family=quasipoisson()` as the response distribution. What do you notice about the predicted abundance?
- Can you work out a way to use `ldply()` from the `plyr` package so that you can use `facet_wrap` in `ggplot2` to plot predictions for multiple models in a grid layout?