Spatial Statistics

Computer lab – Session 3

Madlene Nussbaum

15 Oct 2024

Table of contents

1 Spatial prediction of binary response by random forest

1.1 Explore data

We will spatially predict landslide incidents in dataset `lsl` from R package `spDataLarge` (see excercieses *session 1*). The response is a binary variable `lslpts`. Further, five covariates derived from the elevation model are available.

> Task 1
>
> Investigate the dataset: Compute the percentage of each class. Is the dataset balanced?

Task 2

Create boxplots of response vs. covariates. Which relationships do you see?

1.2 Fit random forest model

Task 1

Fit a random forest model using the terrain information as covariates (without spatial coordinates `x` and `y`).
You can for example use package `ranger`. Use arguments `importance = "permutation"` to obtain permuted covariate importance and `probability = T` to obtain probability predictions for the next tasks.

Task 2

Evaluate covariate importance using the function `importance()`. Create a barplot for better visibility. Is the performance of the covariates as expected from the exploratory analysis?

1.3 Compute predictions and covariate interpretation

Load the area to be predicted containing the covariates:

```
library(terra)
ta <- rast(system.file("raster/ta.tif", package = "spDataLarge"))
```

Task 1

Create predictions for the entire extent of the `ta.tif`. Use `predict()` with the random forest object from the section above. Use `as.data.frame()` to transform the raster values into a `data.frame`. What information/values does the function `predict()` return?

Task 2

Create a `SpatRaster` object with the function `rast()` containing the predictions for the class `TRUE`. Plot the resulting map.

Hint

You can get `x` and `y` coordinates for each pixel using `as.data.frame(, xy = TRUE)`.

The predictions were now created for the entire extent. We only have observations in the central part. Therefore, we need crop the predictions by the study area to avoid spatial extrapolation.

Note: this could also have been done before predicting to increase computational efficiency.

Load the study area:

```
data("study_mask", package = "spDataLarge")
```

> Task 3
>
> Set the pixels outside of the study area to `NA` using the function `mask()` from `terra`.

1.4 Model interpretation (optional)

Inform yourself on accumulated local effects plots: Chapter 8.2, Molnar, 2024.

> Task 1
>
> Print accumulated local effects plots for all covariates using the function `ALEPlot()` from the package `ALEPlot`. What is the interpretation of these plots?
> Note: You will need to define a prediction function (`yhat` on the help page of `ALEPlot()`). Make sure this function returns a vector of the probabilities just for one class, see output of `predict()` above.

1.5 Spatial coordinates as covariates

```
a <- 30 # rotation angle
x*cos(a/180*pi) - y*sin(a/180*pi)
x*sin(a/180*pi) + y*cos(a/180*pi)
```

> Task 1
>
> Add spatial coordinate axis including rotation by e.g. 30 and 60 degrees as additional covariates. Fit the above random forest model again and compute predictions and create the map.

Task 2

Compare the models by comparing the out-of-bag (OOB) model fit, covariate importance and the output maps. Create a difference map with `new.raster <- raster1 - raster2`.
If you have done so above, also compute accumulated local effects plots.

2 Spatial predict continuous response with random forest

We will use the `berne` soil mapping dataset from the R package `geoGAM`. Due to size limitations of CRAN `berne.grid` for predictions only covers a very small part of the study area. We can still use it, to exemplify the prediction process.

Use topsoil pH `ph.0.10` as response. Starting from column `cl_mt_etap_pe` you find a large number of potential covariates in the dataset.

```
library(geoGAM)
data(berne)
data(berne.grid)
```

2.1 Explore dataset

Task 1

Plot a histogram of the response. Investigate the value range and the completeness of the data.

Task 2

Screen the covariates. What data types are there?

Task 3

Create an `sf` object and display the data points with the response using `mapview`. The coordinate reference system information can be found on the help page of the dataset.

2.2 Fit random forest model

Task 1

Fit a random forest model using `ranger`. Remove rows with missing values in the response or the covariates. We use the predefined calibration and validation datasets. Hence, only use the rows labeled "calibration" in row `dataset`.

Task 2

Run the Boruta model selection algorithm with function `Boruta()` from the R package `Boruta`. Remove covariates that are considered unimportant according to the results.

Task 3 (optional)

Tune the number of covariates randomly chosen for testing at each tree split (`mtry`). You can either implement this yourself (`apply` or `for` over the possible `mtry` values which are `1:number.covariates`, then select `mtry` of the model with the lowest OOB mean squared error) or use a meta-package like `caret`.

Hint

Here some input how it can be done with `caret`:

```
# create a matrix with all settings to test
# we limit ourself to mtry
options.to.test <- expand.grid(
  .splitrule = c("variance"),
  .min.node.size = c(5),
  .mtry = c(5, 15, 20, ... )
)
# model training with different settings
rf.caret <- train(
  x = d.ph[, names .. of covariates still in the game ],
  y = d.ph$ph.0.10,
  tuneGrid = options.to.test,
  method = "ranger",
  trControl = trainControl(method = "cv", number = 10)
)
```

2.3 Investigate spatial structure of residuals

Task 1

Compute the random forest residuals (use `predict()` on the calibration data). Investigate the residuals for spatial structure. Plot the residuals against `x` and `y` axis and create a sample variogram.

Task 2

Test if the model can be improved by adding rotated coordinate axis. Compare the OOB mean squared error of the model with and without the coordinates.

2.4 Create predictions

Task 1

Compute predictions for `berne.grid` and plot the result. Are you happy with the visual result?

Task 2

Compute predictions for the rows labeled with "validation" in the column `dataset` in the `berne` dataset. Compute the R^2 and compare it with the OOB R^2.

Task 3

Save the predictions for the "validation" rows together with the observed values and coordinates. We will inspect them in the next lab session.