PE&RC Postgraduate course

"Uncertainty Analysis and Statistical Validation of Spatial Environmental Models"

Monday 9-12-2024: Uncertainty propagation in vegetation indices by 1st order Taylor series approximation

Introduction

In this course we will use two different methods for analysing input uncertainty propagation through spatial models. This first practical involves the 1st order Taylor series method, which approximates the spatial model by a linear function for which analytical results can be obtained.

We start by applying the 1st order Taylor series method on two commonly used vegetation indices to compute how uncertainties in input reflectances in different spectral bands propagate to uncertainty in the indices. The aim of the exercises is to get acquainted with the method and to assess whether the method is suitable for the model(s) you use in your own research. The data provided in this practical are intended for demonstration purposes only and they should not be used outside the context of the exercises.

Most computations will be performed in the R language and environment for statistical computing and graphics. R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It installs and runs on a wide variety of UNIX platforms and similar systems (including Linux), Windows and MacOS (see: http://www.r-project.org/).

R is extended via packages. There are several packages supplied with the R distribution and many more are available through the CRAN* family of internet sites covering a very wide range of modern statistics (R Development Core Team, 2024). We will use some packages that are particularly suitable for analysing spatial data with R.

Vegetation indices

A spectral vegetation index is generated by combining reflection data from multiple spectral bands (e.g. from satellite imagery) into a single value. Spectral vegetation indices are designed to enhance the vegetation signal in remotely sensed data (typically form visible red and near-infrared regions) for assessing the amount of live, green vegetation. Such assessments are used for many applications such as forest and land degradation studies and precision agriculture. Reflectance of green leaves in the visible part of the spectrum is very low, with a slightly higher bump in the green. In the near-infrared, the reflectance of green leaves is much greater than in any portion of the visible spectrum. Other materials such

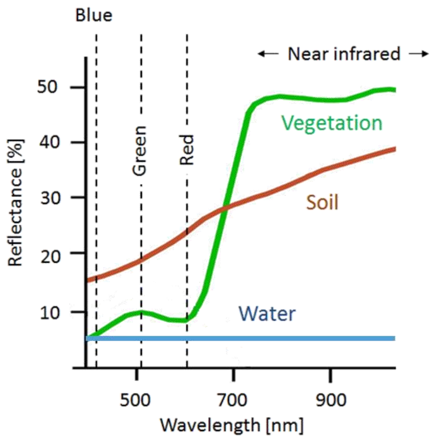

Fig. 1. Typical spectral signatures of vegetation, (dry) soil and water.

* CRAN: Comprehensive R Archive Network

as bare soil, sand, exposed rock and concrete generally show a steadier rise in reflectance as wavelength increases from the visible to the near-infrared, whereas deep water tends to absorb most incoming radiation (see Fig. 1). Two often used vegetation indices are the Simple Ratio (SR) and the Normalized Difference Vegetation Index (NDVI):

SR = NIR / RED

NDVI = (NIR - RED) / (NIR + RED)

where NIR denotes the reflectance fraction in the near-infrared spectral band and RED is the reflectance in the visible red spectral band. Figure 2 shows the Simple Ratio for the study area used in today's practical. The brighter the image tone, the higher the amount of live vegetative cover. Black to dark grey areas correspond to bare soil, water bodies and roads.

Reflectance measurements are subject to measurement error caused by sensor noise, atmospheric conditions as well as other factors. The standard deviations of the measurement errors in the two spectral bands are indicated by σ_{NIR} and σ_{RED}, respectively. Atmospheric conditions and the corrections thereof contribute to correlation of the measurement errors in the two bands; this correlation is indicated by correlation coefficient ρ.

Fig. 2. Simple Ratio for the study area.

First order Taylor approximation

Measurement errors in NIR and RED reflectances propagate to the vegetation indices, which leads to erroneous assessment of vegetative coverage. Equation 7 in the chapter "Propagation of error in spatial modelling with GIS" (Heuvelink, 1999; see link in references) describes the general 1st order Taylor series method:

$$\tau^2 \approx \sum_{i=1}^{m}\sum_{j=1}^{m} \rho_{ij}\sigma_i\sigma_j g_i'(\bar{b})\, g_j'(\bar{b})$$

where τ^2 is the variance of the output of an operation $g(.)$ on m input variables, $\bar{b} = (b_1, \dots, b_m)$ represents the central values of the input and g_i' is the first derivative of $g(.)$ with respect to its *i*-th argument (partial derivative with respect to input variable *i*).

Exercise 1: *Show that using this method the variance of SR is approximated by:*

$$\tau_{SR}^2 \approx \frac{\sigma_{NIR}^2}{RED^2} + \sigma_{RED}^2 \cdot \frac{NIR^2}{RED^4} - 2\rho \cdot \sigma_{NIR} \cdot \sigma_{RED} \cdot \frac{NIR}{RED^3}$$

What is the effect of ρ on uncertainty about SR?

Exercise 2: *Try to derive a similar formula for τ_{NDVI}^2 by yourself. The formula will be discussed in due time. What is the effect of ρ on uncertainty about NDVI? Help for computing derivatives can be found at:* http://www.mathsisfun.com/calculus/derivatives-rules.html.

If you can't wait for the plenary feedback session of the answer, you can click [here](#) for a sneak preview.

Software and data

R and RStudio™ (an integrated development environment for R) are assumed to have been installed on your computer. If not, you can install R after downloading it from the R website http://cran.r-project.org/index.html. RStudio can be found at. If you are new to R, you may want to study the brief introduction: http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf.

When you start RStudio, you should see an interface composed of several windows, as in Figure 3 (without the contents).

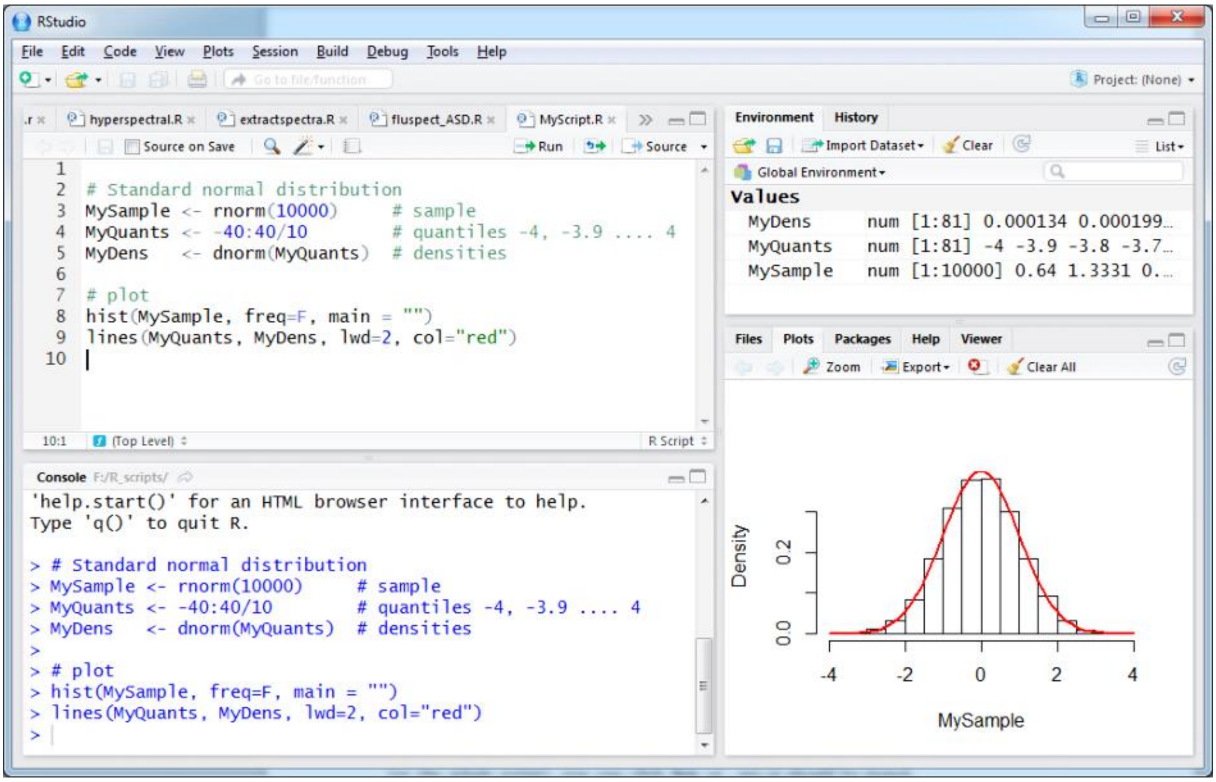

Figure 3. Interface of RStudio with editor window (top left), console window (bottom left), workspace/history (top right) and files / plots / packages / help (bottom right).

You will have to install a few packages for handling and analysing spatial data; you can do this as follows:

- Start RSudio
- Select the Packages tab in the bottom right window
- Click on Install
- In the Packages field type: In the Packages field type: `sf, gstat, ranger, spatstat, terra, TUWmodel`

Installation of packages typically needs to be done only once. However, occasionally it happens that packages are removed by mistake. In that case, you will receive an error message when attempting to load the library (for example: `there is no package called 'sf'`) and you will have to repeat the above steps.

The data files for this practical are in the file FCLorraine.zip, folder General > Course Materials > Monday > Practical. Store the contents in a (new) folder on your computer, e.g., D:\ PERC\Lorraine.

R commands can be typed in the R console (bottom left panel in Fig. 3), but it is more convenient to type and store them in a script (top left panel). You can create a new script via `File | New File | R script`. **Save your script regularly!**

Help on an R function (e.g. sqrt) can be found by typing `help(sqrt)` or `?sqrt` on the R prompt.

Make functions in R

The following lines of R code first define a function for computing τ_{SR} and next apply that function on some input data.

Exercise 3: *Verify that the function TaylorSR implements the formula given in Exercise 1. Enter (copy) the code in a script file (in RStudio). Run the script by selecting the code and pressing Ctrl-Enter (if using the RStudio interface).*

```
# Uncertainty SR by Taylor series method, returns tau.
# nir and red are the measured reflectances in the two
# bands; s_red and s_nir are the standard deviations
# of the measurement errors; rho is the correlation
# between the two measurement errors.

TaylorSR <- function(red, nir, s_red, s_nir, rho) {
 tau_sq <- s_nir^2/red^2 +
 s_red^2*nir^2/red^4-2*rho*s_nir*s_red*nir/red^3
 sqrt(tau_sq) # return tau
}

# call the function with some data
TaylorSR(0.1, 0.6, 0.025, 0.03, 0.8)
```

The hatch symbol (#) on the first line specifies a comment on the remainder of a line, which is not being processed by the interpreter.

The standard deviations used in the code above are for demonstration purposes only. We will use these values later in this lab as well:

$$\sigma_{RED} = 0.025, \quad \sigma_{NIR} = 0.03\,, \quad \rho = 0.8.$$

Empirical values for these variables can be obtained from repeated spectral measurements of reference reflectance panels which are used for image calibration, for example. The values for σ_{RED}, σ_{NIR} and ρ are likely to depend on spatially distributed properties such as topography and land cover. However, this is not taken into account in the current lab.

Exercise 4: *Write and run a new function* TaylorNDVI *for the square root of* τ^2_{NDVI} *which you derived in Exercise 2. You will have to avoid computed* τ^2_{NDVI} *values smaller than 0 (which are impossible), for example using the function* `replace`.

Don't give up too easily. Feedback will be provided in due time, but if you get stuck and have no idea how to proceed, you can consult this code snippet.

Apply formulas or functions on series of data

The functions just defined can also be applied to series of data. We will do this to graphically assess how uncertainty about vegetation indices depends on the input data.

Exercise 5: *Add the lines in the below box to your script file and study the code as well as the results of running it. Repeat the calculations for* $\rho = 0.8$. *Do the plots conform to your expectations? Save the plots* (File | Save as | Png).

You are reminded that help on functions can be obtained by typing `help(foo)` or `?foo` on the R prompt (e.g. `?contour` or `help(contour)`). The function `outer` creates a matrix with elements obtained by applying a function specified by `FUN` on its first two inputs. The lower part of the code below exemplifies how one can display multiple plots within a single graphics window and it also illustrates some options for plot embellishment.

```
red <- 1:500/500 # generate a "red" sequence 0.002, 0.004, ..., 1.00
nir <- 1:500/500 # ditto for the nir band

# compute SR for every combination of red and nir using the function outer
SR <- outer(red, nir, FUN = function(r, n) n/r)

# compute uncertainty propagation in SR with fixed s_nir and s_red
tau_SR <- outer(red, nir, TaylorSR, s_red=0.02, s_nir=0.03, rho=0)
CV_SR <- (tau_SR / SR) # coefficient of variation, see Wikipedia

# Open graphics window
x11(width=15, height=5)

# Split window to display multiple plots
par(mfcol=c(1,3),mar=c(4,4,2,1), mai=rep(0.6,4), xaxs="i", yaxs="i", cex=1)

# contour plots with specified levels
contour(z=SR, levels=c(0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 50),
        xlab="RED", ylab="NIR", main="SR", labcex=1)
contour(z=tau_SR, levels=c(0.025, 0.05, 0.1, 0.5, 1, 5, 50),
        xlab="RED", ylab="NIR", main="Tau(SR) by Taylor method", labcex=1)
contour(z=CV_SR, levels=c(0.02, 0.05, 0.1, 0.2, 0.5, 1, 2), labcex=1,
        xlab="RED", ylab="NIR", main="CV(SR) by Taylor method")
```

Exercise 6: *Make (and save) similar plots for the uncertainty on NDVI given the same ranges of reflectance. Try to understand differences in the coefficients of variation of NDVI and SR. Only if needed you can consult this code snippet.*

Apply functions on a remotely sensed image

Next, we apply the functions on remotely sensed imagery. The study area (Fig. 2) is located in the Belgian Lorraine region (49°41'50"N 5°28'31"E) and is characterised by sandy-loam and loamy-sand soils with strong ferric components. This site has a mean altitude of 350 m.a.s.l. with a rather flat topography, a mean annual temperature of 8.5ºC and an annual precipitation of 1013 mm. The main land cover types are arable crops, grass and trees. You may want to take a look at: https://www.google.nl/maps/@49.6943,5.4783,2808m/data=!3m1!1e3.

A flight campaign with the AHS-160 sensor (http://www.argonst.com/products.htm; http://en.openei.org/wiki/AHS160(formerly_Daedalus)) took place on June 20, 2005 under clear sky and dry soil surface conditions. This sensor provided a total of 63 spectral bands covering the visible (VIS: 430-700 nm), near infrared (NIR: 700-1100 nm) and Short-Wave infrared (SWIR: 1100-2540 nm) parts of the electromagnetic spectrum. The data were corrected for geographical, radiometric and atmospheric attenuations by the Central Data Processing Center of the Vlaamse Instelling voor Technologisch Onderzoek (CDPC-VITO, Belgium). The image was post-processed to obtain the following false colour bands at 2.5 m spatial resolution:

Layer	Wavelength name	Centre wavelength [nm]
1	Green	560
2	Red	661
3	NIR	835

Exercise 7: *Close any remaining open graphics device windows. Run the code below and study the functions used. Note that default graphical parameters are saved and later restored to circumvent a potential issue with changed settings after running* `plotRGB`. *Interpret the histograms. Are they consistent with the above table and the typical spectral signature of vegetation? What seems to be the unit (if any) of the data?*

```
library(terra)
setwd("D:/PERC/FCLorraine") # set to the folder where downloaded data are
false_color <- rast("FCLorraine.tif")  # reads in a SpatRaster

# close any separate R graphical devices
while (!is.null(dev.list())) dev.off()

op <- par(no.readonly = TRUE) # save default graphical settings
plotRGB(false_color, 3, 2, 1, stretch='lin')
par(op) # restore default graphical settings

summary(false_color, maxsamp=2e5)
hist(false_color, 1, maxcell=2e5)
hist(false_color, 2, maxcell=2e5)
hist(false_color, 3, maxcell=2e5)
```

New SpatRasters with SR and tau_SR data can be computed and the result can be displayed as shown below. The function `app` of the terra package is a powerful function similar to the function `apply` of R's base package for computing a new SpatRaster from another SpatRaster using a function, such as the function TaylorSR. Here we put a wrapper around TaylorSR to deal with the specific arguments used by the function.

```
SRim <- app(false_color, fun=function(x) x[3]/x[2], filename="SR.tif",
             overwrite = T)

plot(SRim, col=gray(1:100/100), main="SR")

tau_SR <- app(false_color, fun=function(x)
  TaylorSR(x[2], x[3], 0.025, 0.03, 0.8), filename="tau_SR.tif",
  overwrite = T)

plot(tau_SR, col=gray(1:100/100), main= "Tau(SR)")
```

Exercise 8: *Compute and display the coefficient of variation for SR. Note: You can combine two (or more) SpatRasters using the function* `c`*, after which* `app` *can be used for computing the coefficient of variation.*

Again, don't give up too easily, but if you get stuck and have no idea how to proceed, you can consult this code snippet

Exercise 9: *Compute raster layers for NDVI, tau_NDVI, and CV(NDVI) and inspect the results. Compare CV(NDVI) and CV(SR).*
Can you explain the bright white spot in the upper right of the CV(NDVI) map? Save the plots (File | Save as | Png).

Finally

Don't forget to save your script! Tomorrow, you will compare results obtained with the 1st order Taylor approximation method with results from the Monte Carlo method.

References

Heuvelink, G.B.M., 1999. Propagation of error in spatial modelling with GIS. In: Geographical Information Systems. Volume 1, 2nd Edition. Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W. (Eds). John Wiley & Sons, Inc. pp. 207-217 (link).

R Development Core Team, 2024. An Introduction to R - Notes on R (https://cran.r-project.org/doc/manuals/r-release/R-intro.html).