Allier dataset and model

Kasia Sawicka

May 2016

Quick tutorial for Allier data

Introduction

The moisture content at wilting point is an important input data for the crop WOFOST model. Becasue the moisture content varies considerably over study area in a way that is not linked directly with soil type, it was necessary to map its variation separately to see how moisture limitations affect the calculated crop yield.

Unfortunately, becasue moisture content must be measured on samples in the laboratory, it is expensive and time consuming to determine it for a sufficiently large number of data points for creating the prediction maps by kriging. An alternative and cheaper way is to calculate it from other indicators. Because the moisture content at wilting point is often strongly correlated with moisture content at field capacity and the soil porosity. both of which can be measured more easily, it was decided to investigate how errors in measuring and mapping these would work through to a map of calculated moisture content. Calculation of the moisture content can be done using a pedo-transfer function, which in this case takes the form of multiple linear regression (see model function).

Data and model preparation

1. Define environmental model as a function in R. For example, model for the moisture content at wilting point Θ_{wp}:

$$\Theta_{wp}^{'} = \beta_0 + \beta_1 * \Theta_{fc} + \beta_2 * \Phi + \epsilon + \delta$$

where: $\Theta_{wp}^{'}$ denotes measured moisture content at wilting point, β_0, β_1 and β_2 are the model parameters, δ is residuals resulting from measurment error lack of model fit.

```
allierModel <- function(field.capasity, soil.porosity, beta0, beta1, beta2, delta) {

  soil.moisture <- beta0 + beta1 * field.capasity + beta2 * soil.porosity + delta
  soil.moisture

}
```

2. Create spatial simulations of a gstat object.

```
library(sp)
library(rgdal)
```

```
## rgdal: version: 1.1-8, (SVN revision 616)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.0.1, released 2015/09/15
##  Path to GDAL shared files: d:/R/lib/rgdal/gdal
##  GDAL does not use iconv for recoding strings.
```

```
##  Loaded PROJ.4 runtime: Rel. 4.9.1, 04 March 2015, [PJ_VERSION: 491]
##  Path to PROJ.4 shared files: d:/R/lib/rgdal/proj
##  Linking to sp version: 1.2-3
```

```r
library(gstat)

Allier <-  read.csv("Allier.csv", header=TRUE)
coordinates(Allier) = ~x+y

# prediction grid Allier in local co-ordinate system
Allier_grd <- readGDAL("AllierMask.asc")
```

```
## AllierMask.asc has GDAL driver AAIGrid
## and has 84 rows and 100 columns
```

```r
# gstat object for cokriging
g <- gstat(id=c("fc"), formula = fc~1, data = Allier,
           model = vgm(0.0027, "Sph", 480, 0.0013), nmax = 24)
g <- gstat(g, id = "por", formula = por~1, data = Allier,
           model = vgm(0.0029, "Sph", 480, 0.0008), nmax = 24)
g <- gstat(g, id = c("fc", "por"),
           model = vgm(0.0013, "Sph", 480, -0.0008))
g <- fit.lmc(variogram(g), g)

# Co-kriging
nsim = 20
set.seed(1234567)
Allier_krig <- predict(g, Allier_grd, nsim = nsim, debug.level = -1)
```

```
## drawing 20 multivariate GLS realisations of beta...
## Linear Model of Coregionalization found. Good.
## [using conditional Gaussian cosimulation]
##
 61% done
100% done
```

```r
str(Allier_krig)
```

```
## Formal class 'SpatialGridDataFrame' [package "sp"] with 4 slots
##   ..@ data       :'data.frame':  8400 obs. of  40 variables:
##   .. ..$ fc.sim1  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim2  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim3  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim4  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim5  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim6  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim7  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim8  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim9  : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim10 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim11 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim12 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
```

```
##   .. ..$ fc.sim13 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim14 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim15 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim16 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim17 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim18 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim19 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ fc.sim20 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim1 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim2 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim3 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim4 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim5 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim6 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim7 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim8 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim9 : num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim10: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim11: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim12: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim13: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim14: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim15: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim16: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim17: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim18: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim19: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ por.sim20: num [1:8400] NA NA NA NA NA NA NA NA NA NA ...
##   ..@ grid       :Formal class 'GridTopology' [package "sp"] with 3 slots
##   .. .. ..@ cellcentre.offset: Named num [1:2] 12.5 12.5
##   .. .. .. ..- attr(*, "names")= chr [1:2] "x" "y"
##   .. .. ..@ cellsize         : num [1:2] 25 25
##   .. .. ..@ cells.dim        : int [1:2] 100 84
##   ..@ bbox       : num [1:2, 1:2] 0 0 2500 2100
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "x" "y"
##   .. .. ..$ : chr [1:2] "min" "max"
##   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
##   .. .. ..@ projargs: chr NA
```

3. Create simulations of model parameters.

```
# for simplicity assume all normally distributed
b0 <- rnorm(n = nsim, mean = -0.26, sd = 0.1)
b1 <- rnorm(n = nsim, mean = 0.41, sd = 0.2)
b2 <- rnorm(n = nsim, mean = 0.49, sd = 0.25)
delta <- rnorm(n = nsim, mean = 1E-4, sd = 1E-5)
```

Then use these realizations for error propagation with propagate() from ‘spup’.