Assess congruence between patchy mosaics using vectorial Kappa (κ_v) in R

Vincent Bonhomme

May 6, 2014

1 Preliminaries

This vignette intends to be a very short user manual to calculate vectorial Kappa (or κ_v) using the KappaV package in R.

It assumes a basic preliminary knowledge in R language. Hopefully, many ressources are available on the CRAN's homepage[1]. KappaV must of course be installed and loaded.If that's not the case, the two lines below will do the job:

```
# you will only have to install package once
install.packages("KappaV")
```

```
# but loading it is mandatory for every new R session
require(KappaV)
```

Many approaches have been developped to compare two rasters, including the Cohen's Kappa index. A simple approach would be to rasterize the vectorial mosaics but it is neither elegant nor exact.

We[2], extended the Cohen's Kappa index to directly use the vectorial mosaics as input data leading to the filling of a confusion matrix, and then to the κ_v value. The basic idead behind is to intersect polygons of the two mosaics, calculate areas of the resulting polygons and use them to fill the confusion matrix.

2 Calculate vectorial Kappa

I will use below two shapefiles that can be downloaded from my website there: `http://www.vincentbonhomme.fr/KappaV/shp.zip`.

This archive contains two shapefiles created with Voronoi tesselation using the DYPAL platform. Be sure to change the paths according to the location of your own shapefiles. We will use the path to the `.shp` files, and the companion files (`.dbf`, `.prj`,etc.). On my systems these paths are `" /Desktop/shp/voro"` and `" /Desktop/shp/voro2.shp"`, respectively.

You will also need to specify, for each shapefile, which colums in your `.dbf` file contain the polygon IDs and the nominal variable of interest. If these informations are contained in columns which names are `ID` and `OD`, then you do not have to worry about that, these are the default values. For instance, in the example below, you can omit the last line. Then, the κ_v calculation is straightforward:

```
kv <- KappaV("~/Desktop/shp/voro2.shp",
             "~/Desktop/shp/voro2.shp",
             "ID", "ID", "OS", "OS")

       0      1      2      3      4
0      0 184436 230630 151246 300794
1  88549      0      0      0      0
2 130008      0      0      0      0
3  87361      0      0      0      0
4 180221      0      0      0      0
```

[1] http://cran.r-project.org/

[2] Myself, Mathieu Castets, Jules Morel and Cédric Gaucherel. I worked myself at the French Institute of Pondicherry, India when I wrote this package. I'm presently affiliated both to the School of Mathematics and Statistics, Sheffield, UK and to the UMR CBAE, Montpellier, France.

```
kv

$confusion.matrix
       0      1      2      3      4
0      0 184436 230630 151246 300794
1  88549      0      0      0      0
2 130008      0      0      0      0
3  87361      0      0      0      0
4 180221      0      0      0      0

$kappa.v
   Kappa  Kappa.sd
0 -0.413 0.0003003
```

If you have any trouble with this package, have a look to `?KappaV`, then feel free to contact me: `bonhomme.vincent@gmail.com`.