

## VARIOGRAMAS, KRIGING

### 1 Objetivo

Disponemos de una simulación de un campo aleatorio en un cuadrado de  $[0; 100] \times [0; 100]$ , o sea 10201 puntos. Los datos están en el archivo *simulation.txt*, en una tabla de 3 columnas : las coordenadas  $x$ ,  $y$  y el valor  $z$  del campo en estas coordenadas. Vamos a sortear unos 100 datos de la tabla (un dato es una fila de la tabla) y con ellos estimar un variograma, ajustar-lo con varios modelos, construir los mapas según los variogramas con el kriging y comparar-los entre ellos y a la simulación original.

Utilizaremos el software R, con los paquetes *geoR* et *fields*.

### 2 Sorteo aleatorio

Utilizar la función *sample* para constituir una muestra

```
IndMuestra = sample(1:10201,100)
Muestra = Simu[IndMuestra,]
```

luego se ha de convertir esta muestra en datos geoestadísticos para ser utilizados por *geoR*.

```
datos = as.geodata(Muestra)
plot(datos)
```

### 3 Variograma

#### 1. Variograma empírico :

La función *variog* calcula el variograma empírico. Las diferentes opciones permiten variar el número de clases, sus amplitudes, la distancia máxima, ... La función *plot(objeto variog)* traza el resultado. Variando las opciones, elegir un ajuste.

```
vario.c = variog(datos,op="cloud") # nube variografica
plot(vario.c,main = "Nube variografica",pch='+')
vario.b = variog(datos) # variograma empirico sin opcion
plot(vario.b,main = "Variograma empirico")
vario.b = variog(datos,max.dist = ? ,breaks=?) # argumento distancia maxima y
# ajuste de clase
plot(vario.b,main = "Variograma empirico")
```

#### 2. Variograma ajustado :

La función *variofit* ajusta un variograma admisible al variograma empírico. Se pueden elegir varios modelos (exponencial, gaussiano, esférico, ...), la presencia o no de una pepita, los valores iniciales de los parámetros para el algoritmo de optimización ... La función *lines(objeto variofit)* traza el variograma ajustado en un gráfico ya existente, habitualmente el variograma empírico.

```
varioest = variofit(vario.b)
plot(vario.b, main="modele exponential")
lines(varioest)
```

Vamos a comparar los diferentes ajustes obtenidos con varios modelos y con la presencia o no de una pepita.

```
varioest_i = variofit(vario.b,cov.model = ?, ini.cov.pars=?, fix.nugget = TRUE, nugget= ?)
lines(varioest_i)
```

## 4 Kriging

La funcion *krige.conv* produce un mapa obtenido con kriging con el variograma elegido en cada punto de una grilla. Tambien produce la varianza de kriging para cada sitio. Los parametros (variograma, tipo de de kriging) estan definidos con la funcion *krige.control*.

La grilla se puede construir con la funcion *expand.grid*.

La funcion *image(objeto kriging)* traza el resultado del kriging.

Realizar los mapas de kriging con los diferentes variogramas y comentar los resultados. El paquete *fields* sirve para cambiar la paleta de colores. Para añadir una leyenda se necesita ampliar el cuadro de diseño.

```
grilla = expand.grid(0:100,0:100)
Kcontrol = krige.control(type.krige = "ok", obj.model = varioest_i)
K = krige.conv(datos, loc = grilla, krige = Kcontrol)

image(K,col = tim.colors(64), ylim = c(-10,100), zlim = c(-5,5),
      x.leg = c(0,100), y.leg = c(-10,-5) ,coords.data = datos$coords)
title(main="Kriging Modelo 1")

image(K,col = tim.colors(64),ylim = c(-10,100),zlim = ?,
      val = sqrt(K$krige.var),x.leg = c(0,100),y.leg=c(-10,-5), coords.data = datos$coords)
title(main = "Desviacion estandar de kriging Modelo i")

image(K,col = tim.colors(64), ylim = c(-10,100), zlim = c(-5,5) ,val = Simu$z,
      x.leg = c(0,100), y.leg = c(-10,-5), coords.data = datos$coords)
title(main = "Simulacion inicial")
```

## 5 Simulaciones condicionales

La funcion *output.control(n.pred=n,simul=TRUE,thres=umbral)* permite realizar simulaciones condicionales y a partir de estas calcular probabilidades de no sobrepasar un umbral. Se deduce entonces un mapa de probabilidad de sobrepaso.

```
grilla2 = expand.grid(seq(0,100,2),seq(0,100,2))# grilla fina para acelerar el calculo.
Ocontrol = output.control(n.pred = 100,simul = TRUE, thres = ?)
Ks = krige.conv(datos, loc= grilla2, krige = Kcontrol, out = Ocontrol)

image(Ks,val = Ks$simulations[,?], col = tim.colors(64), ylim = c(-10,100), zlim = c(-5,5),
      x.leg = c(0,100), y.leg = c(-10,-5), coords.data = datos$coords)
title(main="Simulacion condicionnal")

image(Ks,val = 1-Ks$probabilities.simulations,col =tim.colors(64), ylim=c(-10,100),
      zlim = ? , x.leg = c(0,100), y.leg = c(-10,-5), coords.data = datos$coords)
title(main="Probabilidad sobrepaso umbral ?")
```