Laboratorio3

Mariana Cubero

24/9/2020

Este Laboratorio es un ejercicio para evaluar diferentes métodos y configuraciones para hacer interpolación con datos puntuales a un área de interés. Además se analizan las limitaciones que se pueden presentar al trabajar con datos reales.

```
rmsenn <- rep(NA, 5)
for (k in 1:5) {
  test <- dta[kf == k, ]
  train <- dta[kf != k, ]
  gscv <- gstat(formula=prec~1, locations=train, nmax=5, set=list(idp = 0))
  p <- predict(gscv, test)$var1.pred
  rmsenn[k] <- RMSE(test$prec, p)
}
```

```
## Warning in proj4string(d$data): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## [inverse distance weighted interpolation]
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved
```

```
## Warning in proj4string(d$data): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## [inverse distance weighted interpolation]
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved
```

```
## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## [inverse distance weighted interpolation]

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## [inverse distance weighted interpolation]

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## [inverse distance weighted interpolation]

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved

## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
rmsenn

## [1] 200.6222 190.8336 180.3833 169.9658 237.9067
```

```
## [1] 200.6222 190.8336 180.3833 169.9658 237.9067
mean(rmsenn)

## [1] 195.9423

## [1] 195.9423
1 - (mean(rmsenn) / null)

## [1] 0.5498908

## [1] 0.5498908
```

Question 1: Describe what each step in the code chunk above does

Paso 1 es cargar los datos y darles una breve "ojeada" o exploración a la variable de interés que es la precipitación anual.

Paso 2 Es dar el formato espacial necesario para aplicar los modelos del ejercicio y visualizar la distribución de los puntos.

Paso 3 Ajustar el sistema de coordenadas para que los mapas que se vayan a usar calzen adecuadamente.

Paso 4 Crear una función que calcule el RSME para evaluar los modelos.

Paso 5 Se dibujan los polígonos de proximidad que crean áreas alrededor de los puntos observados, entonces en zonas con mayor cobertura de puntos los polígonos son más pequeños y caso contrario lo polígonos más grandes se crean debido a la mayoy distancia entre puntos observados.

Paso 6 Intersecar los polígonos con el mapa para cortar sólo los polígonos que están dentro de los límites del estado que se está estudiando.

Paso 7 Se hace la interpolación con con voronoi y se compara el RMSE con un modelo nulo usando validación cruzada.

Question 2: How does the proximity-polygon approach compare to the NULL model?

Los polygonos de proximindad lo que hacen es como partir la zona de interés en secciones más pequeñas según la cantidad de observaciones puntuales que haya agrupadas y en cada polýgono se iterpola el nivel de lluvias a áreas más grandes. En el caso del RMSEA que compara ambos abordajes considerando en el caso del modelo nulo solo valores de lluvia anual de las observaciones puntuales y en el caso de los poligonos considera los promedios de cada área ya considerando la interpolación.

Question 3: You would not typically use proximty polygons for rainfall data. For what kind of data would you use them?

Sería más útil para datos que presenten una cobertura de área más uniforme pueden brindar mejores resultados en la generación de los polygonos. y que sea consistente en el tiempo.

Question 4: IDW generated rasters tend to have a noticeable artefact. What is that?

IDW en este caso tiende a sobreestimar la incidencia de las precipitaciones en las zonas donde hay menos información puntual disponible.

```
rmse <- rep(NA, 5)
for (k in 1:5) {
  test <- dta[kf == k, ]
  train <- dta[kf != k, ]
  gs <- gstat(formula=prec~1, locations=train)
  p <- predict(gs, test)
  rmse[k] <- RMSE(test$prec, p$var1.pred)
}
```

```
## Warning in proj4string(d$data): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## [inverse distance weighted interpolation]
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved
```

```
## Warning in proj4string(d$data): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## [inverse distance weighted interpolation]
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved
```

```
## Warning in proj4string(d$data): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## [inverse distance weighted interpolation]
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## [inverse distance weighted interpolation]

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## [inverse distance weighted interpolation]

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"):
Discarded datum Unknown based on GRS80 ellipsoid in CRS definition,
##  but +towgs84= values preserved

## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
rmse

## [1] 215.3319 211.9383 190.0231 211.8308 230.1893

## [1] 215.3319 211.9383 190.0231 211.8308 230.1893
mean(rmse)

## [1] 211.8627

## [1] 211.8627
1 - (mean(rmse) / null)
```

```
## [1] 0.5133192
```

[1] 0.5133192

Question 5: Inspect the arguments used for and make a map of the IDW model below. What other name could you give to this method (IDW with these parameters)? Why?

Este modelo considera el vecino más cercano u observación para realizar la interpolación. Y al fijar el idp a 1 le asigna todo el peso al mismo, por lo que le llamaría modelo nulo con el vecino más cercano o modelo estimado con un sólo predictor. Está como "fijando" la estimación al vecino más cercano.

```
library(dismo)
library(gstat)
nfolds <- 5
k <- kfold(aq, nfolds)
ensrmse <- tpsrmse <- krigrmse <- idwrmse <- rep(NA, 5)
for (i in 1:nfolds) {
  test <- aq[k!=i,]
  train <- aq[k==i,]
  m <- gstat(formula=OZDLYAV~1, locations=train, nmax=opt$par[1],
set=list(idp=opt$par[2]))
  p1 <- predict(m, newdata=test, debug.level=0)$var1.pred
  idwrmse[i] <-  RMSE(test$OZDLYAV, p1)
  m <- gstat(formula=OZDLYAV~1, locations=train, model=fve)
  p2 <- predict(m, newdata=test, debug.level=0)$var1.pred
  krigrmse[i] <-  RMSE(test$OZDLYAV, p2)
  m <- fields::Tps(coordinates(train), train$OZDLYAV)
  p3 <- predict(m, coordinates(test))
  tpsrmse[i] <-  RMSE(test$OZDLYAV, p3)
  w <- c(idwrmse[i], krigrmse[i], tpsrmse[i])
  weights <- w / sum(w)
  ensemble <- p1 * weights[1] + p2 * weights[2] + p3 * weights[3]
  ensrmse[i] <-  RMSE(test$OZDLYAV, ensemble)
}
```

```
## Warning in proj4string(d$data): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(d$data): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output
```

```
## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output
```

```
rmi <- mean(idwrmse)
rmk <- mean(krigrmse)
rmt <- mean(tpsrmse)
rms <- c(rmi, rmt, rmk)
rms
```

```
## [1] 8.041305 8.307235 7.930799
```

```
## [1] 8.041305 8.307235 7.930799
rme <- mean(ensrmse)
rme
```

```
## [1] 7.858051
```

```
## [1] 7.858051
```

Question 6: Which method performed best?

El método “thin plate spline” es ligeramente mejor que el IDW, el modelo con mayor RSME es el krigin ordinario. Se llega a esta conclusión considerando la validación cruzada que compara el RSME ponderado, sin embargo el modelo ensamblado a partir de la ponderación del RMSE es el que da mejores resultados.

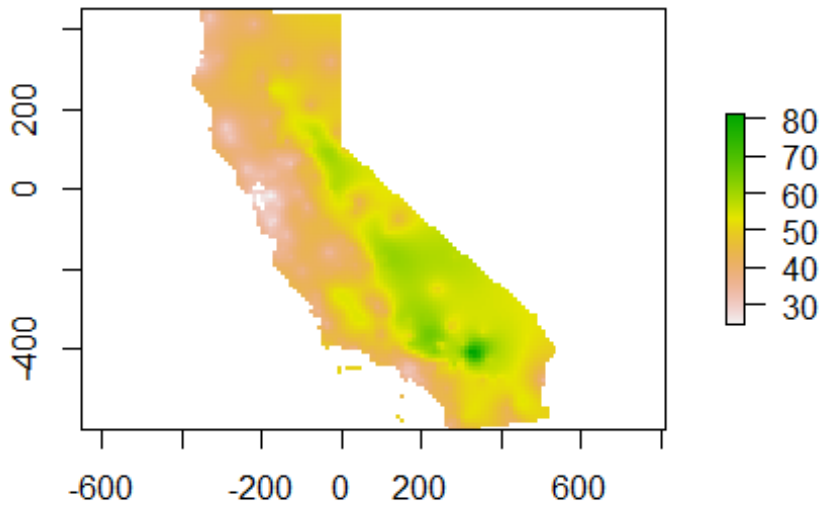
Question 7: Show where the largest difference exist between IDW and OK.

La principal diferencia entre estos dos modelos es que el IDW tiende a interpolar puntos con valores más altos de contaminación en comparación el OK que presenta un patrón de

contaminación ligeramente más concentrado en ciertas áreas. Este último patrón muestra más similitudes con demás modelos (TPS y Ensemble).

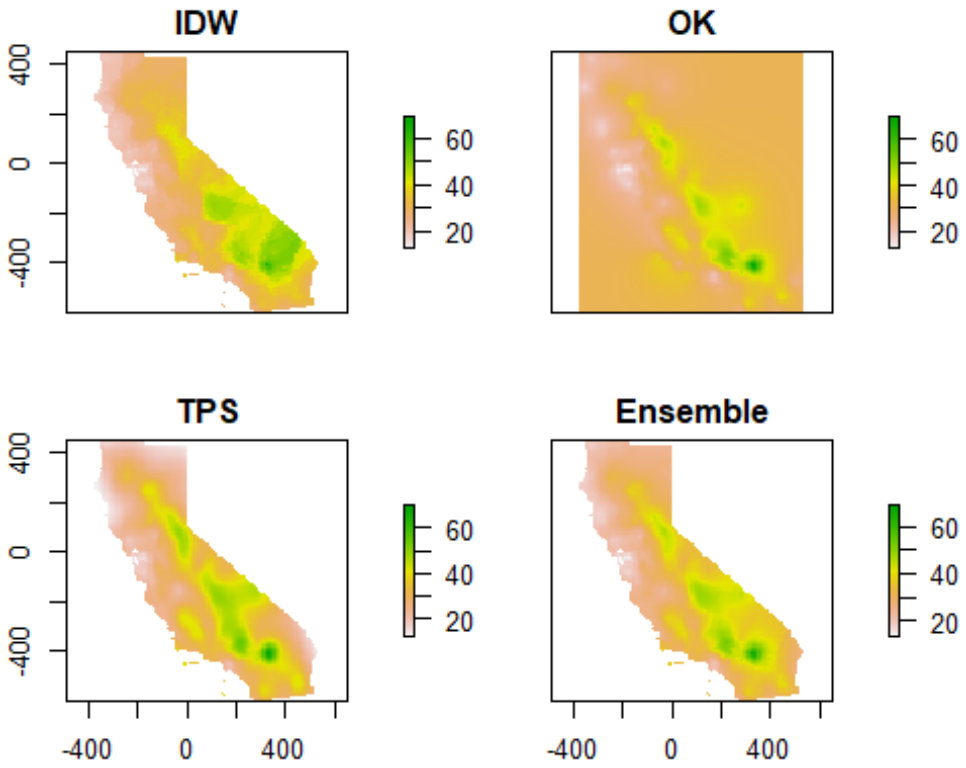
Question 8: Show where the difference between IDW and OK is within the 95% confidence limit of the OK prediction.

```
k <- gstat(formula=OZDLYAV~1, locations=aq, model=fve)

# predicted values
kp <- predict(k, g)

## Warning in proj4string(d$data): CRS object has comment, which is lost in
output

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

## [using ordinary kriging]

## Warning in proj4string(newdata): CRS object has comment, which is lost in
output

ok <- brick(kp)
#se forman los intervalos de confianza
r<-brick(ok$var1.pred) + sqrt(brick(ok$var1.var))*1.96
ok1 <- mask(r, ca)
plot(ok1)
```



El patrón observado en este intervalo de confianza es similar al que se tiene con el modelo IDW. Donde las zonas con valores más altos se extienden más allá del área central del mapa.

```
s <- stack(idw, ok[[1]], tps, ensemble)
names(s) <- c('IDW', 'OK', 'TPS', 'Ensemble')
plot(s)
```



Question 9: Can you describe the pattern we are seeing, and speculate about what is causing it?

Tres de los 4 modelos que se están comparando tienen patrones similares en las áreas centrales del mapa. Sin embargo, en la parte norte, donde de paso hay menos puntos para hacer la interpolación se estiman valores más bajos. Mi especulación en este caso es que si bien hay zonas que tienen condiciones que favorecen la concentración de contaminación por distintas razones, la distancia y cantidad de puntos para hacer interpolación está afectando las estimaciones en las zonas más alejadas del centro.