

LatticeKrigDemo.R

nychka

2023-07-10

```
#####  
### Lattice vs spatialProcess demo  
#####
```

```
library(LatticeKrig)
```

```
## Loading required package: spam
```

```
## Spam version 2.9-1 (2022-08-07) is loaded.  
## Type 'help( Spam)' or 'demo( spam)' for a short introduction  
## and overview of this package.  
## Help for individual functions is also obtained by adding the  
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##  
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':  
##  
##      backsolve, forwardsolve
```

```
## Loading required package: fields
```

```
## Loading required package: viridisLite
```

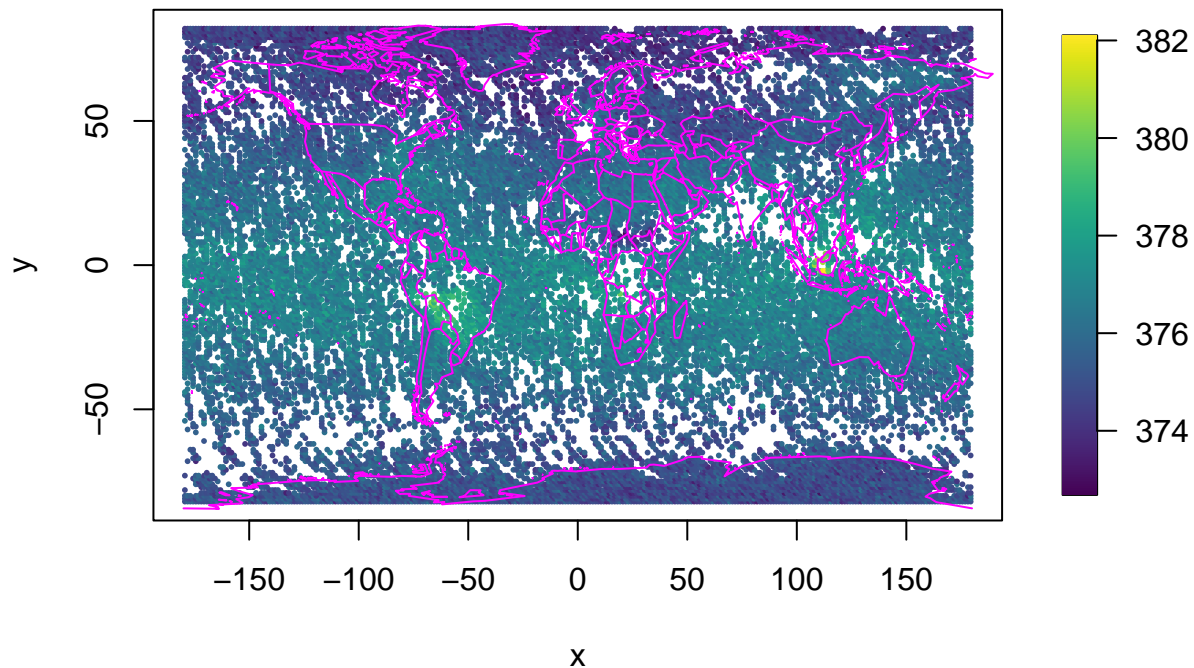
```
##  
## Try help(fields) to get started.
```

```
data(CO2)
```

```
s<- CO2$lon.lat  
z<- CO2$y  
dim( s)
```

```
## [1] 26633      2
```

```
bubblePlot( s,z, highlight=FALSE, size=.4)  
world( add=TRUE, col="magenta")
```



```
# 921 locations
ind1<-which(
  s[,1]>= -110 & s[,1] <= -60 &
  s[,2]>= 20 & s[,2] <= 55
)

# 1884 locations
ind2<- which(
  s[,1]>= -120 & s[,1] <= -50 &
  s[,2]>= 0 & s[,2] <= 55
)

# 3420 locations
ind3<- which(
  s[,1]>= -120 & s[,1] <= -50 &
  s[,2]>= -55 & s[,2] <= 55
)

system.time(
  fit1<- spatialProcess(s[ind1,], z[ind1], cov.function="Tps.cov" )
)
```

```
##      user  system elapsed
##    1.993    0.171    2.164
```

```
# for 961 locations get about 2 seconds
system.time(
  fit2<- spatialProcess(s[ind2,], z[ind2], cov.function="Tps.cov" )
)
```

```
##      user  system elapsed
##    5.884    0.737    6.623
```

```
# for n=1884 get about 7 seconds
```

```
system.time(  
  fit3<- spatialProcess(s[ind3,], z[ind3], cov.function="Tps.cov" )  
)
```

```
##      user  system elapsed  
## 27.888   3.553  31.447
```

```
# for n=3420 get about 35 seconds  
# increasing n by 2 => ~factor of 5  
# O(n^3) complexity implies factor of 8
```

```
# approx thin plate spline fit using fixed rank Kriging  
system.time(  
  fit4<- LatticeKrig(s, z, a.wght = 4.01 )  
)
```

```
##      user  system elapsed  
## 34.171   0.710  34.881
```

```
# for ~27K locations get about 35 seconds
```

```
# summary of fit  
fit4
```

```
## Call:  
## LatticeKrig(x = s, y = z, a.wght = 4.01)  
##  
##  
## Number of Observations:                26633  
## Number of parameters in the fixed component 3  
## Effective degrees of freedom (EDF)      800.2  
## Standard Error of EDF estimate:         7.894  
## MLE sigma                              0.5069  
## MLE rho                                6.296  
## MLE lambda = sigma^2/rho                0.04081  
##  
## Fixed part of model is a polynomial of degree 1 (m-1)  
##  
## Summary of estimated fixed model coefficients  
##           Estimate Std. Error      t value Pr(>|t|)  
## Intercept  3.749934e+02 1.144268973 327.71436617 0.0000000  
## x1        -6.171181e-04 0.006787081  -0.09092541 0.9275525  
## x2        -6.771584e-03 0.009510819  -0.71198746 0.4764788  
## Standard errors are based on generalized LS  
## and for covariance parameters fixed at the estimated values  
##  
## Basis function : Radial  
## Basis function used: WendlandFunction  
## Distance metric: Euclidean
```

```
##
## Lattice summary:
## 3 Level(s) 16050 basis functions with overlap of 2.5 (lattice units)
##
## Level Lattice points Spacing
##      1      1242 10.2500
##      2      3483  5.1250
##      3     11325  2.5625
##
## Nonzero entries in Ridge regression matrix 9848348
```

```
# lower bound for spatialProcess would be about 5^3*33 seconds ~ 60 minutes
# (but more likely 8^3 * 33 4 hours due to O( n^3) complexity)
```

```
surface( fit4)
world( add=TRUE, col="magenta")
```

