R Programs for Spatio-temporal Modeling

Jiaqi Xu

The State Key Laboratory of Information Engineering in Surveying, Mapping,
and Remote Sensing, Wuhan University
Luoyu Road 129, Wuhan, China
Telephone: 13407176820
E-mail: xujiaqi.77@163.com

1. Introduction

Kriging has been implemented efficiently in spatial domain in many existing software, such as ArcGIS, R and so on. However, it has not been implemented in spatio-temporal domain efficiently so far. This paper demonstrates how to extend the implementation of kriging from space domain to spatio-temporal domain using R. We use product-sum model as our space-time variogram because it's easy to compute. In order to save computation time, we remove data beyond range manually according to the graphics of spatial variogram and temporal variogram. To simplify computation, we use the list object in R to store intermediate result.

2. Experimental data sets

Two data sets have been imported into R. One is the rainfall data in XML format in Northeastern China from the 49th day in 2000 to the 353th day in 2005, at a temporal resolution of 16days. The other is radiance data of the 26th row and the 4th column of the MODIS image, which is used for tailoring the scope of kriging. After resampling, the result image has a spatial resolution of 50000m. We use the spatio-temporl rainfall data of sampled sites to predict the rainfall data in the whole region on the 177th day of 2005. The output of the rainfall prediction has the same spatial resolution and the same spatial scope as the MODIS image has.

station ID	x	y	2000_049	2000_065	2000_081	2000_097	2000_113
50442	1383524	5564465	0	0.0625	0.9375	2.9375	17.625
50468	1624582	5599338	0.25	0	0.6875	9.8125	12.8125
50527	1089407	5382676	0.4375	1.125	3.5	8.3125	16.5
50548	1379578	5430511	0.25	1.25	1.0625	1.8125	20.5
50557	1490600	5448995	0	0	1.1875	1.5	10.5
50564	1637448	5511004	0.1875	0	2.4375	6.875	19.25
50618	995234.7	5258546	0.125	0.625	0	1.5	15
50632	1256364	5359822	0.125	0.6875	3.625	1.5625	11.375
50639	1331414	5287694	0.125	1.375	0.625	0	38.125
50656	1604621	5374503	0.0625	0.0625	2.9375	1.625	7.5625
50658	1563492	5339295	0	0	3.8125	0.75	8.3125

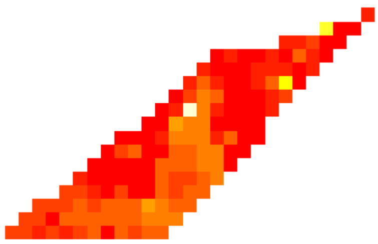

Figure 1. Imported datasets

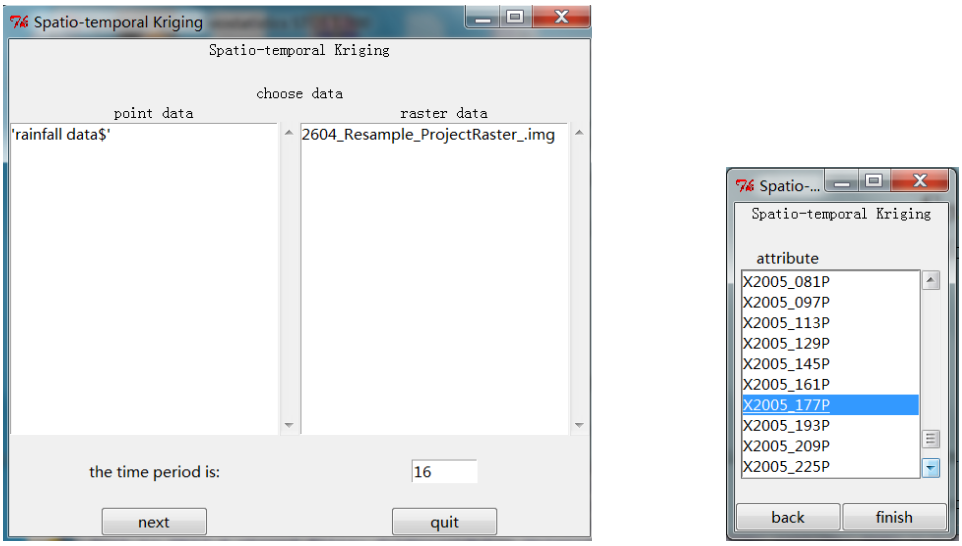

Figure 2. Interface of imported data sets and output results

3. Principles of spatio-temporal modeling

3.1 Product-sum model

The product-sum variogram model, introduced by De Cesare and Myers (2001), can be specified in equation 1:

$$\gamma_{s,t}(h_s, h_t) = \gamma_{s,t}(h_s, 0) + \gamma_{s,t}(0, h_t) - k\gamma_{s,t}(h_s, 0)\gamma(0, h_t) \qquad (1)$$

It is known that this theoretical model of variogram is easily fitted with the use of “marginal” variograms (De Iaco and Myers, 2001)(purely spatial variogram and purely temporal variogram). In nature, this product-sum model has formalized spatiotemporal dependency, so that it can be used for not only estimating values at unobserved locations but also prediction in future time. The product-sum model is potentially useful for analyzing air pollution data, meteorological data, ground water data and so on. In this paper we have implemented the product-sum variogram model for spatio-temporal kriging of the rainfall data in the MODIS-image-specified region on the 177th day of 2005. Some R functions in package gstat on CRAN are used.

3.2 Spatio-temporal ordinary kriging

Spatio-temporal kriging has the same principle of interpolation as the spatial kriging has, that is, best, linear, unbiased estimation(BLUE). Ordinary kriging filters the mean from the simple krigng estimator by requiring that the kriging weights sum to one (Deutsch and Journel, 1998). This results in the following ordinary kriging estimator:

$$Z^*(x_0) = \sum_{i=1}^{n} \lambda_i Z(x_i) \qquad (2)$$

while the standard error of predicted value is given in equation 3 (Zhang 2005).

$$SE = \sqrt{\sum_{i=1}^{n} \lambda_i \gamma(x_0 - x_i) + \mu} \qquad (3)$$

$$\begin{vmatrix} \gamma_{11} & \cdots & \gamma_{1n} & 1 \\ \vdots & & \vdots & \vdots \\ \gamma_{n1} & \cdots & \gamma_{nn} & 1 \\ 1 & \cdots & 1 & 0 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \mu \end{vmatrix} = \begin{vmatrix} \gamma_{01} \\ \vdots \\ \gamma_{0n} \\ 1 \end{vmatrix} \quad (4)$$

On the left side of the equation 4, γ_{ij} stands for the spatio-temporal variogram between two known points, while on the right side of the equation, γ_{ij} for the spatio-temporal variogram between a known point and a predicted point. λ_i are prediction weights to be solved and μ is the Lagrange multiplier.

4. Computation of spatio-temporal modeling

4.1 Determination of parameter k in product-sum model

Using sample data, first we calculate each empirical spatial variogram at different time points. The function variogram() returns a list with three useful elements, that is: the first column np stands for the number of points within a certain lag; the second column dist stands for spatial lags and the last column gamma stands for empirical values of spatial variograms corresponding to spatial lags in the second column. The first element in var1 has been shown below the following R codes.

```
> dataframe1 <- list()
> var1 <- list()
> for(i in 1:n1)
+ {
+     dataframe1[[i]] <- data.frame(x,y,sdata[i])
+     coordinates(dataframe1[[i]]) = ~x+y
+     var1[[i]] <- variogram(sdata[,i]~1,dataframe1[[i]])
+ }
```

```
> var1
[[1]]
    np      dist    gamma dir.hor dir.ver   id
1    6  48849.95 110.1735       0       0 var1
2  109  93972.27 127.8544       0       0 var1
3  160 144348.17 146.4464       0       0 var1
4  204 197642.96 201.7778       0       0 var1
5  254 255176.69 218.3722       0       0 var1
6  254 310790.26 239.4228       0       0 var1
7  271 368137.50 261.4314       0       0 var1
8  298 424448.12 250.3777       0       0 var1
9  281 481249.57 336.0209       0       0 var1
10 281 536625.84 282.5770       0       0 var1
11 296 596224.97 303.1030       0       0 var1
12 261 652680.70 367.3214       0       0 var1
13 235 708098.52 372.5503       0       0 var1
14 240 767237.22 294.9834       0       0 var1
15 220 821859.40 349.4586       0       0 var1
```

After transposing the original data matrix, we set vector with zero elements as x coordinates, and time span as y coordinates and compute each empirical temporal variograms at different spatial locations. The first element in the list var2 has been shown below the following R codes.

```
> data<-dataset[[select.index1]]
> tdata <- t(as.data.frame(data[,-(1:3)]))
> x <- matrix(0,nr=nrow(tdata),nc=1)
> y <- matrix(seq(from=1,by=period,length.out=nrow(tdata)),nr=nrow(tdata),ncol=1)
> n2 <- ncol(tdata)
> dataframe2 <- list()
> var2 <- list()
> for(i in 1:n2)
+ {
+     dataframe2[[i]] <- data.frame(x,y,tdata[,i])
+     coordinates(dataframe2[[i]]) = ~x+y
+     var2[[i]] <- variogram(tdata[,i]~1,dataframe2[[i]])
+ }
```

```
> var2
[[1]]
    np      dist    gamma dir.hor dir.ver   id
1  267  23.97004 232.1699       0       0 var1
2  393  63.91858 407.1000       0       0 var1
3  384 111.91667 545.7562       0       0 var1
4  375 159.91467 615.2011       0       0 var1
5  366 207.91257 609.8338       0       0 var1
6  357 255.91036 523.8890       0       0 var1
7  348 303.90805 372.9220       0       0 var1
8  339 351.90560 252.8740       0       0 var1
9  330 399.90303 308.6362       0       0 var1
10 321 447.90031 486.1552       0       0 var1
11 312 495.89744 634.4249       0       0 var1
12 303 543.89439 607.0589       0       0 var1
13 294 591.89116 585.6011       0       0 var1
14 285 639.88772 506.6862       0       0 var1
15 276 687.88406 355.1576       0       0 var1
```

Following that, we can get all the spatial varigrams and temporal variograms. However, there is only one spatial varigram and one temporal variogram are used to represent the whole region. So we averaged all the spatial variograms with the same spatial lag and averaged all the temporal variograms with the same temporal lag. The algorithm is programmed with the following R codes.

```
> nrow1 <- nrow(var1[[1]])
> gamma_s <- numeric(nrow1)
> for(i in 1:nrow1)
+ {
+     sum <- 0
+     for(j in 1:n1)
+     {
+         sum <-  sum + var1[[j]]$gamma[i]
+     }
+     gamma_s[i] <- sum/n1
+ }
> var1[[1]]$gamma <- gamma_s
> svar <- var1[[1]]
```

```
> nrow2 <- nrow(var2[[1]])
> gamma_t <- numeric(nrow2)
> for(i in 1:nrow2)
+ {
+     sum <- 0
+     for(j in 1:n2)
+     {
+         sum <- sum + var2[[j]]$gamma[i]
+     }
+     gamma_t[i] <- sum/n2
+ }
> var2[[1]]$gamma <- gamma_t
> tvar <- var2[[1]]
```

Finally, the parameters k of product-sum model are determined. k must be less than or equal to $\frac{1}{\max\{k_s C_s(0), k_t C_t(0)\}}$ (S.De Iaco, 2001). In our case, the upper bound equals to 0.0001638 caculated with the sample data. In this paper we take the value of k as 0.0001 to ensure that the k value just varies within the range. The most appropriate k value can be evaluated with the effective results of spatio-temporal kriging. The spatio-temporal random field Z(S,T) is assumed to be intrinsic stationary, thus the value of parameter k can be used in the whole region.

4.2 Data selection

Although theoretically there exists screen effect and relay effect(Chilès and Delfiner (1999)), removing data beyond range is a simple and practical way for data selection in kriging, such as in the Spatial Analyst module in ArcGIS. The computation time, drastically increases with the number of data retained, approximately in proportion to

[n]3(Goovaerts (1997)).

The first thing we should do in kriging step is to exclude data beyond range. This algorithm enables the user to remove data beyond range manually according to the graphics of spatial variogram and temporal variogram given in the product-sum model. There is no need to remove seasonal trends as (De Cesare and Myers 2002) does, because from the graphics below we can see that the temporal range is about 100 days, within which there doesn't exist any seasonal trends.

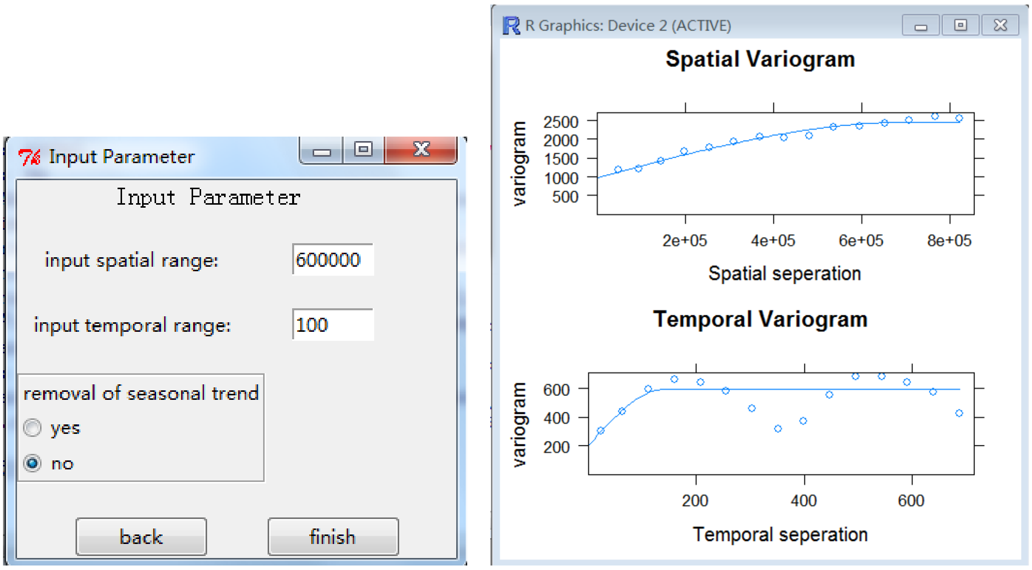

Figure 3. Remove data beyond range according to the graphics

```
> datatemp <-  data[,((stk.var-3)-(nt-1)):(stk.var-3)]
> data1 <- data.frame(x,y,datatemp)
> stdata <- list()
> for(i in 1:npre)
+     stdata[[i]] <- data1
> for(j in 1:npre)
+ {
+     n_delete <- 0
+     for(i in 1:n2)
+     {
+         if(distance[i,j] >= sr){
+             stdata[[j]] <- stdata[[j]][-(i-n_delete),]
+             n_delete <- n_delete+1}
+     }
+ }
```

4.3 Computation of spatio-temporal variograms

After obtaining valid data within spatial range and temporal range, the following critical step is to rearrange the data in order to get matrices with which we compute the spatio-temporal variograms γ_{ij}. For each prediction location, let ns stand for the number of spatial points retained to participate in the kriging computation and nt for temporal points. In the rearranged matrix, there are ns*nt rows and each row stands for a spaio-temporal point. The first column of the matrix stands for the x coordinate, the second column for the y coordinate, the third column for the time information, and the last column for the corresponding data.

```
> krigedata <- list()
> for(i in 1:npre)
+ {
+     ns <- length(stdata[[i]][,1])
+     krigedata[[i]] <- matrix(0,nrow=ns*nt,4)
+     krigedata[[i]][,1] <- rep(stdata[[i]][,1],nt)
+     krigedata[[i]][,2] <- rep(stdata[[i]][,2],nt)
+     for(m in 1:nt)
+         for(n in 1:ns)
+             krigedata[[i]][,3][(m-1)*ns+n] <- (m-1)*period+1
+     krigedata[[i]][,4] <- as.vector(as.matrix(stdata[[i]][,-(1:2)]))
+ }
```

```
>  krigedata[[4]]
         [,1]    [,2] [,3]    [,4]
 [1,] 1909061 5344434    1  2.5625
 [2,] 2034355 5363217    1  2.1250
 [3,] 1920756 5284762    1  6.0000
 [4,] 2076596 5269907    1  1.7500
 [5,] 1909061 5344434   17  0.0000
 [6,] 2034355 5363217   17  0.0000
 [7,] 1920756 5284762   17  0.0625
 [8,] 2076596 5269907   17  0.0000
 [9,] 1909061 5344434   33  2.0625
[10,] 2034355 5363217   33  4.1875
[11,] 1920756 5284762   33  2.6875
[12,] 2076596 5269907   33  2.1250
[13,] 1909061 5344434   49  0.3125
[14,] 2034355 5363217   49  0.6875
```

The spatial distance between two spatio-temporal points i and j is computed with the first column and second column in the rearranged data list krigedata(). And the temporal distance between two spatio-temporal points i and j is determined by the third column in the rearranged data list krigedata(). We use the list distpre() to record spatial distances and temporal distances between the predicted points and the known points that participate in kriging computation.

```
> distpre <- list()
> tnow <- period*(nt-1)+1
> for(i in 1:npre)
+ {
+     ns <- length(krigedata[[i]][,1])
+     distpre[[i]] <- matrix(0,nrow=ns,ncol=2)
+     distpre[[i]][,1] <- sqrt((xpre[i]-krigedata[[i]][,1])^2+(ypre[i]-krigedata[[i]][,2])^2)
+     distpre[[i]][,2] <- tnow - krigedata[[i]][,3]
+ }
> sdistknown <- list()
> tdistknown <- list()
> for(i in 1:npre)
+ {
+     ns <- length(krigedata[[i]][,1])
+     sdistknown[[i]] <- matrix(0,nrow=ns,ncol=ns)
+     for(m in 1:ns)
+       for(n in 1:ns)
+         sdistknown[[i]][m,n] <- sqrt((krigedata[[i]][,1][m]-krigedata[[i]][,1][n])^2+
+         (krigedata[[i]][,2][m]-krigedata[[i]][,2][n])^2)
+     tdistknown[[i]] <- matrix(0,nrow=ns,ncol=ns)
+     for(m in 1:ns)
+         for(n in 1:ns)
+             tdistknown[[i]][m,n] <- abs(krigedata[[i]][,3][m]-krigedata[[i]][,3][n])
+ }
```

Through product-sum model, we compute spatio-temporal variograms γ_{ij} in equation 4 by making use of the distance information stored in the lists (distpre(), sdistknown() and tdistknown()) mentioned above. Here we use the spherical model to fit the empirical variograms.

```
for(i in 1:npre)
{
    ns <- length(krigedata[[i]][,1])+1
    stgamma[[i]] <- matrix(0,nrow=ns,ncol=ns)
    sgamma[[i]] <- matrix(0,nrow=ns-1,ncol=ns-1)
    tgamma[[i]] <- matrix(0,nrow=ns-1,ncol=ns-1)
    pregamma[[i]] <- numeric(ns)
    for(m in 1:ns-1)
    for(n in 1:ns-1)
    {
        sgamma[[i]][m,n] <- Cs0+Cs1*(1.5*(sdistknown[[i]][m,n]/sr)-0.5*(sdistknown[[i]][m,n]^3/sr^3))
        tgamma[[i]][m,n] <- Ct0+Ct1*(1.5*(tdistknown[[i]][m,n]/tr)-0.5*(tdistknown[[i]][m,n]^3/tr^3))
        stgamma[[i]][m,n] <- sgamma[[i]][m,n] + tgamma[[i]][m,n] - k*sgamma[[i]][m,n]*tgamma[[i]][m,n]
    }
```

4.4 Kriging prediction results

Click the button "finish" in fig. 3, we can easily obtain the spatio-temporal kriging results and its standard error. The algorithm puts the spaio-temporal variogram lists above into the kriging equation 4 to solve the linear prediction weights λ_i and the Lagrange multiplier μ in equation 2 and 3.

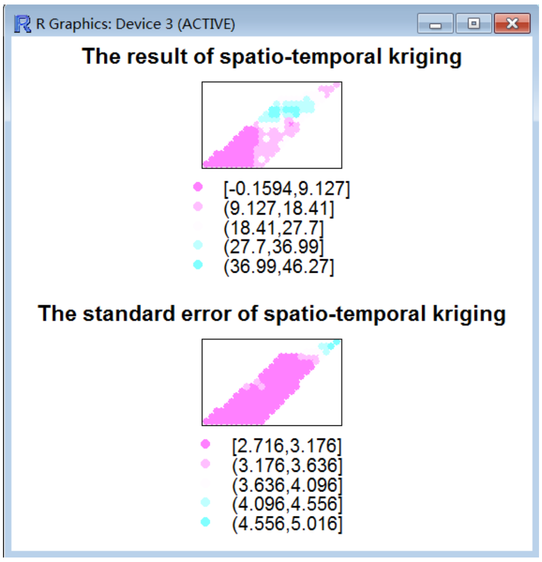

Figure 4. The estimation and standard error of spatio-temporal kriging

5. Conclusion

This paper demonstrates how to implement spatio-temporal kriging using R. The advantage of this method is that it reflects the principle of spatio-temporal kriging. However, there are some works need to be done. This algorithm is not fast enough for practical application. We could parallel it to reduce the execution time of kriging, as well as optimize the algorithm.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.41171313).

7. References

Chilès J. P. and Delfiner P., 1999, *Geostatistics: modeling spatial uncertainty,* New York . Chichester . Weinheim . Brisbane . Singapore . Toronto, A Wiley-Interscience Publication.

De Cesare L., Myers D. E. and Posa D., 2001, Estimating and modeling space-time correlation structures, *Statistics & Probability Letters,* 51: 9-14

De Cesare L., Myers D. E. and Posa D., 2001, Product-sum covariance for space-time modeling: an environmental application, *Environmetrics*. 12:11-23

De Cesare L., Myers D. E. and Posa D., 2002, Fortran programs for space-time modeling, *Computers & Geosciences,* 28:205-212

De Iaco S., Myers D.E. and Posa D., 2001, Space-time analysis using a general product-sum model, *Statistics & Probability Letters,* 52:21-28

Deutsch C. V. and Journel A. G., 1998, *GSLIB: Geostatistical Software Library and User's Guide,* New York, Oxford, Oxford University Press

Goovaerts P., 1997, *Geostatistics for natural resources evaluation,* New York and Oxford, Oxford University Press.

Zhang R. D., 2005, *The theory and application of space variation,* Beijing, Science Press