STAT0017 ICA2

20045458

Question 1 Answers

(1) The four variograms look quite different so we may not assume isotropy, which means the behavior of zinc concentrations in the four directions are different and the trend effect has not been eliminated. All the variograms seem not to have nugget effect expect for the northwest-southeast direction which has a clear intercept on the y-axis. For the 45° direction, the variogram reaches the lowest sill compared to other three directions, which is likely due to significant difference in zinc concentrations in other directions. Besides, in the 90° and 135° directions, the semi-variance begins to decrease when the lag distance exceeds approximately 1000.

(2) The surfaces predicted using anisotropic variograms from spherical, exponential, Matérn ($\nu = 2$) are rough and similar, whilst the surface using anisotropic variograms using Matérn ($\nu = 10$) are way smoother. From the predicted variations, we see that when the direction is about 45°, the variation at this direction is the least, which coincides with variograms above. At the direction of 90° and 135°, the surface sinks at first and then rises, which relates to the corresponding increasing and then decreasing semivariance.

Question 2 Answers

(1) Assume the 2×2 positive definite matrix A is $\begin{pmatrix} a & b \\ b & c \end{pmatrix}$, $\mathbf{h} = (h_1, h_2)^T$, thus

$$A\mathbf{h} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \begin{pmatrix} ah_1 + bh_2 & bh_1 + ch_2 \end{pmatrix}$$

We identify that the semivariogram is in the exponential form with zero nugget effect, so we have

$$\begin{aligned} ||A\mathbf{h}|| &= \sqrt{(ah_1 + bh_2)^2 + (bh_1 + ch_2)^2} \\ &= \sqrt{(a^2 + b^2)h_1^2 + 2b(a + c)h_1h_2 + (b^2 + c^2)h_2^2} \\ &= \sqrt{h_1^2 + \theta_3 h_2^2} \end{aligned}$$

We see that there is no interaction term h_1h_2 so we have $b = 0$ or $a + c = 0$. From inspection we know $a + c \neq 0$ as it requires $\theta_3 = 1$. Therefore, we have $b = 0$, $a = 1$, $c = \sqrt{\theta_3}$, i.e. A $= \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{\theta_3} \end{pmatrix}$. γ_0 is in exponential form where $c_0 = 0$, $c_s = \theta_1$, $a_s = \theta_2$.

(2) As $h_1, h_2 \to \infty$, $\gamma(h) \to \theta_1$ which is the sill, so the effective range is where $\gamma(h)$ reaches $0.95\theta_1$, that is, the exponential term in $\gamma(h)$ reaches 0.05. Given $\theta_3 = 4$, we have the following results:

For N-S direction, we assume that $h_1 = 0$, so

$$e^{-\frac{2h_2}{\theta_2}} = 0.05 \implies h_2 = -\frac{1}{2}\theta_2 \log(0.05)$$

For E-W direction, we assume that $h_2 = 0$, so

$$e^{-\frac{h_1}{\theta_2}} = 0.05 \implies h_1 = -\theta_2 \log(0.05)$$

Thus, the ratio of the effective range in the E-W and N-S directions is

$$\frac{h_1}{h_2} = \frac{-\theta_2 \log(0.05)}{-\frac{1}{2}\theta_2 \log(0.05)} = 2$$

Question 3 Answers

(1) The empirical variogram is as follows:

```
temp <- read.table(file = 'temp.txt', header = T)
dta <- as.geodata(temp)

# Plot the variogram
plot(variog(dta, option='bin'), main='bin')
```

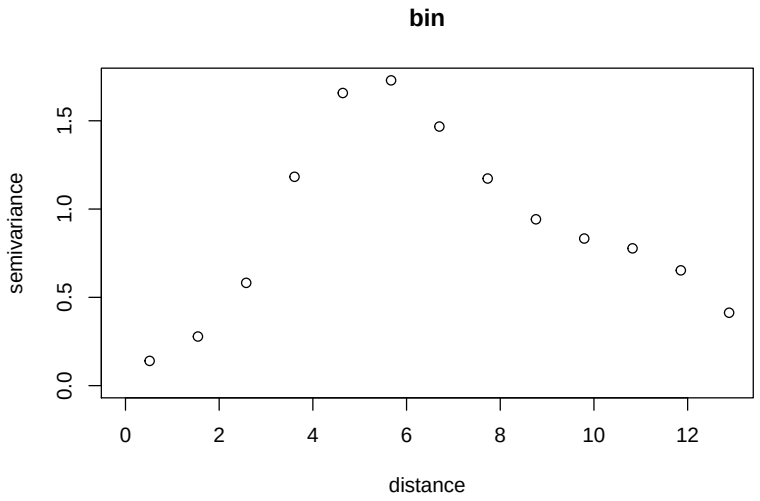

Figure 1: Empirical variogram for the data

(2) When trying to estimate the parameters of a Matérn covariance model using ML or REML, the following error appears:

```
# Error in solve.default(v$varcov, xmat) :
#   system is computationally singular: reciprocal condition number = 7.1299e-22
```

It means that when we were computing the likelihood function, the matrix $A'\Sigma(\theta)A$ is non-invertible mich might be due to some strongly linearly dependent columns.

(3) We use the locations of the data as the region of the mesh because there are plenty of items and there is no need to self-define a domain. We know that the range of `x` is $(137, 148)$ and `y` is $(33, 44)$, so we set the initial `max.edge = c(4, 8)` and decrease it until there is no clearly large triangles in the inner domain (until `max.edge = c(1, 2)`). Also, to avoid some of the triangles with small angles, we increase the `cutoff` until 0.5 so the mesh looks better.

```
coords <- as.matrix(temp[,1:2])
mesh <- inla.mesh.2d(coords, max.edge=c(1, 2), cutoff = 0.5)
```

```
plot(mesh, asp=1)
points(coords, col='red', cex = 0.5)
```

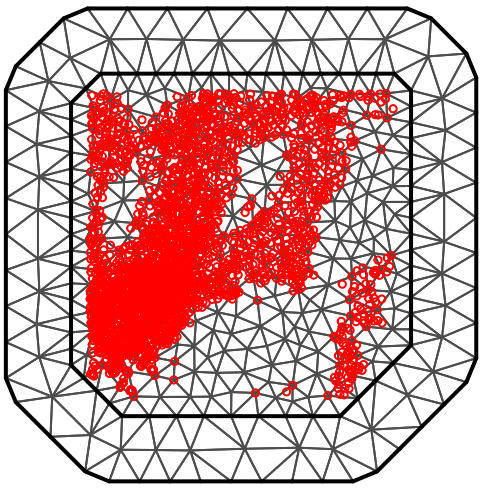

Figure 2: Mesh with the locations of the data points

(4) We carry out a prediction of the data over a grid of dimension 100x100 with plots showing mean and standard deviation of the random field and the response as follows:

```
# Define the projector matrix
A <- inla.spde.make.A(mesh, loc=coords)

spde <- inla.spde2.matern(mesh, alpha=2)

stk <- inla.stack(data=list(resp=temp$t), A=list(A,1),
                  effects=list(i=1:spde$n.spde,
                  m=rep(1,nrow(temp))), tag='est')

res <- inla(resp ~ 0 + m + f(i, model=spde),
             data=inla.stack.data(stk),
             control.predictor=list(A=inla.stack.A(stk)))

# using the default limit covering the domain
pgrid <- inla.mesh.projector(mesh, dims=c(100,100))
prd.m <- inla.mesh.project(pgrid, res$summary.ran$i$mean)
prd.s <- inla.mesh.project(pgrid, res$summary.ran$i$s)

stkgrid <- inla.stack(data=list(resp=NA), A=list(pgrid$proj$A,1),
                      effects=list(i=1:spde$n.spde,
                      m=rep(1,100*100)), tag='prd.gr')

stk.all <- inla.stack(stk, stkgrid)

resg <- inla(resp ~ 0 + m + f(i, model=spde),
              data=inla.stack.data(stk.all),
```

```
                control.predictor=list(A=inla.stack.A(stk.all),
                                       compute=TRUE), quantiles=NULL,
                control.results=list(return.marginals.random=FALSE,
                                     return.marginals.predictor=FALSE))

igr <- inla.stack.index(stk.all, 'prd.gr')$data

grid.arrange(levelplot(prd.m, col.regions=topo.colors(99), main='latent field mean',
                       xlab='', ylab='', scales=list(draw=FALSE)),
             levelplot(matrix(resg$summary.fitt[igr,1], 100),
                       xlab='', ylab='', main='response mean',
                       col.regions=topo.colors(99), scales=list(draw=FALSE)),
             levelplot(prd.s, col.regions=topo.colors(99), main='latent field SD',
                       xlab='', ylab='', scales=list(draw=FALSE)),
             levelplot(matrix(resg$summary.fitt[igr,2], 100),
                       xlab='', ylab='', main='response SD',
                       col.regions=topo.colors(99), scales=list(draw=FALSE)),
             nrow=2)
```

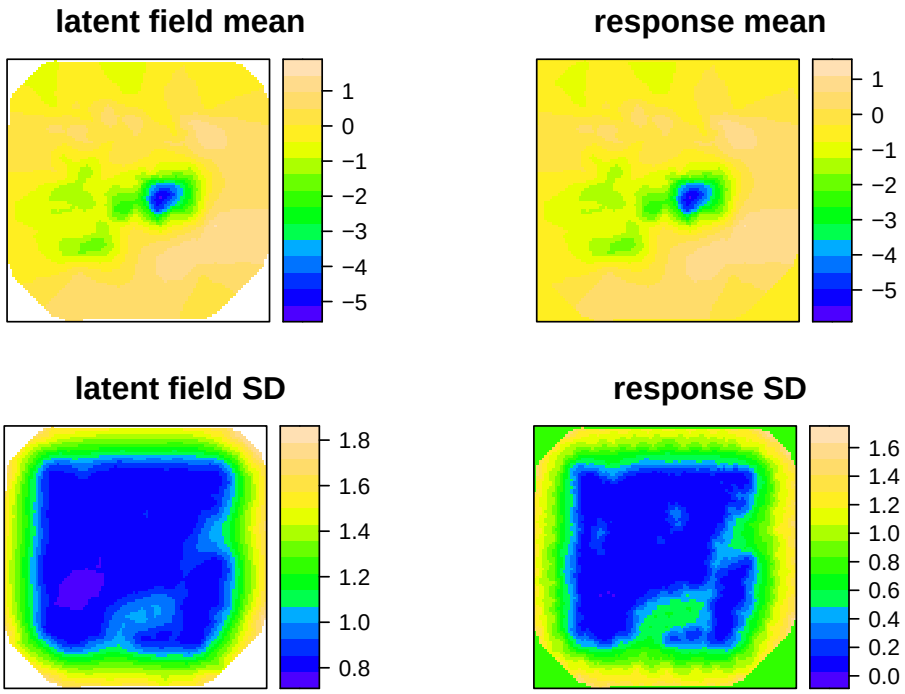

Figure 3: The mean and standard deviation of the random field (top left and bottom left) and the mean and standard variation of the response (top right and bottom right)

The matrices of predicted mean and standard deviation can be accessed using the follwing code:

```
temp.mean <- matrix(resg$summary.fitt[igr,1], 100)
temp.sd <- matrix(resg$summary.fitt[igr,2], 100)
```

(5) The mesh size greatly affected the computational cost of the INLA-SPDE approach. The more nodes on the mesh, the more time needed for computation.

(6) The best choice of mesh size should be the number of items in the dataset. INLA-SPDE can make a better prediction using the whole dataset.