

# Bivariate conditional spatial models: Simulation example in Section 3.2

Noel Cressie and Andrew Zammit-Mangion

Sunday, March 01, 2015

## Setting up

In this vignette we show the *R Software* code used in Section 3.2 of Cressie & Zammit-Mangion (2015). Code for the application study in Section 5 is available in a separate vignette.

In order to run this code, a few packages are needed. The first, **Matrix**, is needed for algebraic operations while **dplyr** and **tidyr** are needed for data manipulation.

```
library(Matrix)
library(dplyr)
library(tidyr)
```

The second set of packages are needed for plotting, and for arranging the figures into panels for publication.

```
library(ggplot2)
library(gridExtra)
library(grid)
library(extrafont)
```

Finally, the package **bicon** provides the machinery for bivariate modelling using the conditional approach with (i) bisquare interaction functions and (ii) Matérn covariance functions for  $C_{11}(\cdot)$  and  $C_{21}(\cdot)$ .

```
library(bicon)
```

We start off by setting up some parameters in the program – these are described in-line.

```
###-----
### Setup
###-----
img_path <- "../paper/art"    ## Where to save the figures
show_figs <- 1                ## Show the figures in document
print_figs <- 0                ## Print figures to file
```

Now we set up the simulation domain. We choose  $D = [-1, 1]$ , and a spacing  $\eta_i = 0.01, i = 1, \dots, 200$ . We collect the grid information in a data frame **df**, to which extra columns will be added further on in the program. We also define **n1** as the number of grid cells for  $Y_1$  and **n2** as the number of grid cells for  $Y_2$ . In this study, **n1** = **n2** = 200 and we define **n** = **n1** + **n2** = 400.

```
###-----
### Construct grid
###-----
ds <- 0.01
df <- data.frame(s=seq(-1+ds/2,1-ds/2,by=ds),
                 areas = ds)
n1 <- n2 <- nrow(df)
n <- n1 + n2
```

Both covariance functions,  $C_{11}(s, u)$  and  $C_{2|1}(s, u)$ , are Matérn covariance functions. That is,

$$C_{11}(s, u) \equiv \frac{\sigma_{11}^2}{2^{\nu_{11}-1}\Gamma(\nu_{11})}(\kappa_{11}|u-s|)^{\nu_{11}}K_{\nu_{11}}(\kappa_{11}|u-s|),$$

$$C_{2|1}(s, u) \equiv \frac{\sigma_{2|1}^2}{2^{\nu_{2|1}-1}\Gamma(\nu_{2|1})}(\kappa_{2|1}|u-s|)^{\nu_{2|1}}K_{\nu_{2|1}}(\kappa_{2|1}|u-s|),$$

where  $\sigma_{11}^2, \sigma_{2|1}^2$  denote the marginal variances,  $\kappa_{11}, \kappa_{2|1}$  are scale parameters,  $\nu_{11}, \nu_{2|1}$  are smoothness parameters, and  $K_\nu$  is the Bessel function of the second kind of order  $\nu$ . The interaction function  $b(s, u)$  is a bisquare function given by

$$b(s, v) \equiv \begin{cases} A\{1 - (|v-s-\Delta|/r)^2\}^2, & |v-s-\Delta| \leq r \\ 0, & \text{otherwise.} \end{cases}$$

In the simulation study we fix  $\nu_{11} = \nu_{2|1} = 1.5$  and set the other parameters (including the standard deviation of the observation error) as follows:

```
###-----
### True process and observation parameters
###-----
kappa1 = 25      ## Scale of C_{11}(. )
kappa21 = 75     ## Scale of C_{2|1}(. )

sigma2_1 <- 1     ## Variance of C_{11}(. )
sigma2_21 <- 0.2  ## Variance of C_{2|1}(. )

A <- 5           ## Amplitude of b(. )
delta = -0.3     ## Shift of b(. )
r = 0.3          ## Aperture of b(. )

sigmav <- 0.5    ## Observation error std
```

## Matrix construction and simulation

After setting the required parameters, we are now in a position to construct the covariance matrix  $\Sigma$ , which is given by

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{11}B^T \\ B\Sigma_{11} & \Sigma_{2|1} + B\Sigma_{11}B^T \end{bmatrix}.$$

To facilitate this we have provided a function `makeSY` which takes a vector of grid distances, the parameters of the Matérn function, and the matrix  $B$  as input arguments. First, we construct the matrix  $B$  that, recall, is simply the interaction function evaluated over the grid cells multiplied by the grid spacing (when using the rectangular rule to approximate the integration). That is,

$$B^{(j,k)} = \eta_k b(s_j, v_k).$$

```
###-----
### Construct required matrices
###-----
H <- t(outer(df$s,df$s,FUN = "-"))      ## Find displacement
B <- A*bisquare_1d(H,delta = delta,r = r)*ds  ## Find Bmat
```

We can now construct the required covariance matrix as follows:

```
D <- abs(H)
Dvec <- as.double(c(D))          ## Find distances
Sigma <- makeSY(r = Dvec,
               var1 = sigma2_1,
               var2 = sigma2_21,
               kappa1 = kappa1,
               kappa2 = kappa21,
               B = B)             ## Build covariance matrix
```

Given the covariance matrix, we can simulate from the bivariate field *jointly*. Observations are simulated from this field by simply adding Gaussian error to the generated fields. These simulations are all added to the data frame df:

```
###-----
### Generate data
###-----
set.seed(50)                      ## Fix seed
samp <- t(chol(Sigma)) %*% rnorm(2*nrow(df)) ## Simulate Y1 and Y2
df <- df %>%
  mutate(samp1 = samp[1:n1],
         samp2 = samp[-(1:n1)],
         Z1 = samp1 + sigmav*rnorm(n1),
         Z2 = samp2 + sigmav*rnorm(n2))    ## Add simulations to df
Z <- matrix(c(df$Z1,df$Z2))           ## Save concatenated observations in Z
```

To demonstrate the benefits of cokriging, we choose to keep only half of the observations of  $Y_1$ , those appearing in the right half of the domain. Inferences on  $Y_1$  in the left half of the domain will be facilitated through observations on  $Y_2$ :

```
keep_Z1 <- 101:200 ## Keep Z1 only in the right half of the domain
keep_Z2 <- 1:200   ## Keep Z2 everywhere
```

## Cokriging

Since we are fixing both processes to have zero mean, cokriging of  $Y_1(s_0)$ ,  $s_0 \in D$  proceeds through the *simple* cokriging equations. These are given through

$$\hat{Y}_1(s_0) \equiv E(Y_1(s_0) \mid Z_1, Z_2) = \begin{bmatrix} c_{11}^T & c_{12}^T \end{bmatrix} \begin{bmatrix} C_{11} + \sigma_{\varepsilon_1}^2 I_{m_1} & C_{12} \\ C_{21} & C_{22} + \sigma_{\varepsilon_2}^2 I_{m_2} \end{bmatrix}^{-1} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix},$$

where for  $q, r = 1, 2$ ,

$$c_{1r}^T \equiv (C_{1r}(s_0, s_{ri}) : i = 1, \dots, m_r); \quad r = 1, 2, \quad (1)$$

$$C_{qr} \equiv (C_{qr}(s_{qi}, s_{rj}) : i = 1, \dots, m_q, j = 1, \dots, m_r); \quad q, r = 1, 2, \quad (2)$$

and  $m_1, m_2$  are the number of observations of  $Y_1, Y_2$ , respectively.

In the following cokriging function, we require four input variables. These are:

- `df`: The original dataframe with information on grid spacings, locations and observations.
- `A`: The amplitude of the bisquare function. If  $A = 0$ , then the two fields are independent.
- `obs_ind`: A vector with values equal to 1 for observations which are kept, and 0 for observations which are omitted.
- `name`: The name to be associated with the cokriging results

The function first constructs the required  $\Sigma$  (through `makeSY`), and then implements the above equations. The function stores the results in the data frame `df`.

```
###-----
### Cokriging function
###-----

co_krige <- function(df,A,obs_ind,name=NULL) {

  B <- A*bisquare_1d(H,delta=delta,r=r)*ds          ## Form B matrix
  Sigma <- makeSY(r = Dvec,
                  var1 = sigma2_1,
                  var2 = sigma2_21,
                  kappa1 = kappa1,
                  kappa2 = kappa21,
                  B = B)
  Zobs <- Z[obs_ind,]                             ## Subset the observations
  mu <- Sigma[,obs_ind] %*%
              solve(Sigma[obs_ind,obs_ind] +
                    sigmav^2 * Imat(length(obs_ind))) %*%
              Zobs                                ## Cokriging equations

  df[paste0(name,"_mu1")] <- mu[1:n1]              ## Save results
  df[paste0(name,"_mu2")] <- mu[-(1:n1)]
  df
}
```

To call the function `co_krige`, we first specify which observations to keep in the variable `obs_ind`:

```
df$keep_Z1 <- 1:nrow(df) %in% keep_Z1    ## Create vector of indices marking which
df$keep_Z2 <- 1:nrow(df) %in% keep_Z2    ## observations are kept and which are discarded
obs_ind <- c(keep_Z1,keep_Z2 + n1)
```

We can then pipe our original `df` through `co_krige` using differing values of `A`: (i)  $A = 0$  (independent variates) and (ii)  $A = A$  (true model). Note that case (i) is identical to simple kriging on  $Y_1$  using only  $Z_1$ , since under independence the system is *autokrigeable* (see Wackernagel, 1995, p. 149).

```
df <- df %>%
  co_krige(A=0,obs_ind = obs_ind,name="ind_model") %>%
  co_krige(A=A,obs_ind = obs_ind,name="true_model")
```

## Plotting

The rest of the code (and the biggest part!) is devoted to plotting. Since this is terse, we do not discuss it in detail. It highly relies on knowledge of the package `ggplot2` and `tidyr`, the latter for putting the data into an appropriate format.

```

###-----
### Plotting
###-----

df_obs <- df %>%
  select(s,Z1,Z2,keep_Z1,keep_Z2) %>%
  gather(obs,z,Z1:Z2) %>%
  filter((keep_Z2 == TRUE & obs == "Z2") | (keep_Z1 == TRUE & obs == "Z1"))

df_estY1 <- df %>%
  select(s,samp1,ind_model_mu1,true_model_mu1) %>%
  gather(process,z,samp1,ind_model_mu1,true_model_mu1)

df_estY2 <- df %>%
  select(s,samp2,ind_model_mu2,true_model_mu2) %>%
  gather(process,z,samp2,ind_model_mu2,true_model_mu2)

obs_plot <- LinePlotTheme() +
  geom_point(data=df_obs,
             aes(x=s,y=z,shape=obs),
             size=3,alpha=1,guide=F) +
  theme(legend.title=element_blank(),
        plot.margin = grid::unit(c(3, 0, 0, 0),units = "mm"))+
  scale_shape_manual(values=c(1,20),guide=F) +
  ylab("")

est_plotY1 <- LinePlotTheme() +
  geom_line(data=df_estY1,
            aes(x=s,y=z,colour=process,linetype=process,size=process)) +
  scale_linetype_manual(values=c("solid","dashed","dotted"),
                        guide=FALSE) +
  scale_size_manual(values=c(0.4,1.3,1.3),guide=F) +
  scale_colour_manual(values=c("black","black","black"),
                      labels=c("Y1",
                                expression(paste(tilde(Y),1)),
                                expression(paste(hat(Y),1))),
                      name="",
                      guide=F) +
  ylab("")

est_plotY2 <- LinePlotTheme() +
  geom_line(data=df_estY2,
            aes(x=s,y=z,colour=process,linetype=process,size=process)) +
  scale_linetype_manual(values=c("solid","dashed","dotted"),
                        guide=FALSE) +
  scale_size_manual(values=c(1,1.3,1.3),guide=F) +
  scale_colour_manual(values=c("black","orange","blue"),
                      labels=c("Y2","IM","TM"),
                      name="") +
  ylab("")

```

```

if(print_figs) ggsave(obs_plot,
                      filename = file.path(img_path,"sim_obs.eps"),
                      width=7,height=4,family="Arial")
if(print_figs) ggsave(est_plotY1,
                      filename = file.path(img_path,"sim_est.eps"),
                      width=7,height=4,family="Arial")
if(show_figs) print(arrangeGrob(obs_plot,est_plotY1,ncol=1),
                    width=16,height=7,family="Arial")

```

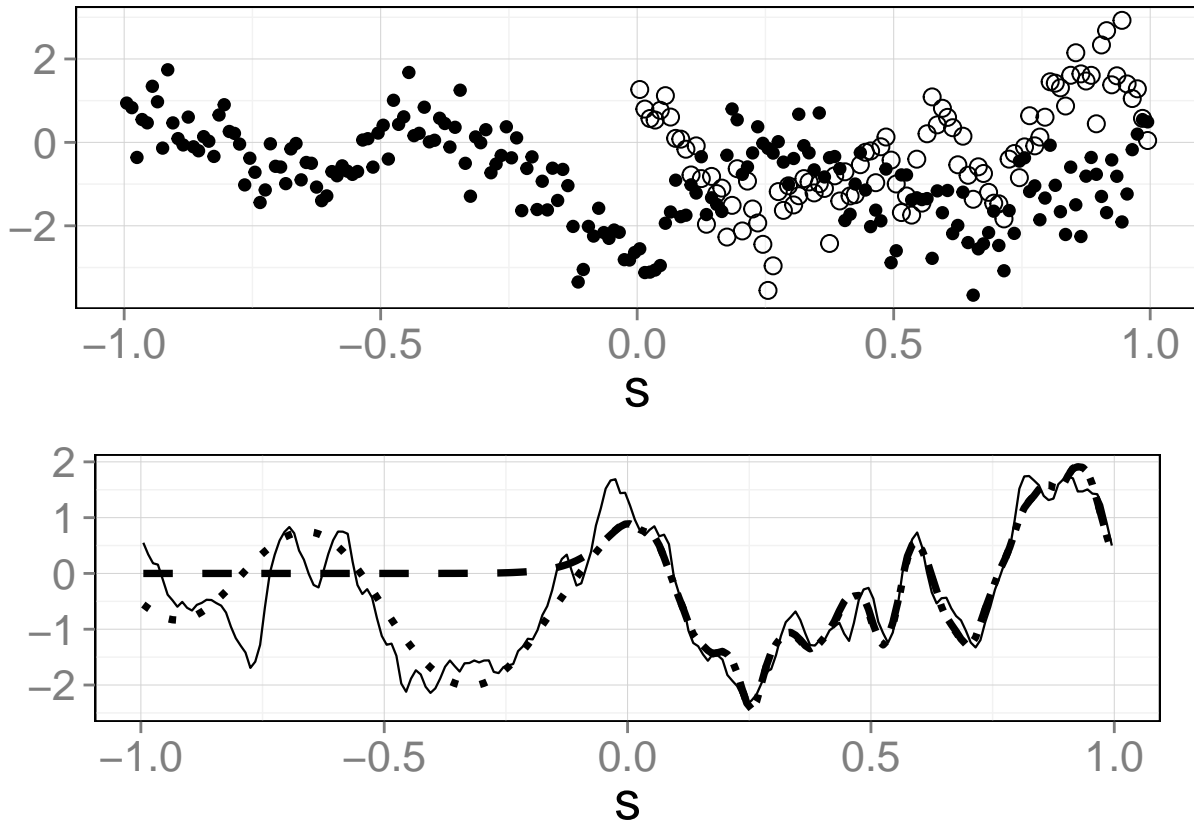


Figure 1: Cokriging using spatial covariances defined by the conditional approach. Top panel: The simulated observations  $Z_1$  (open circles) and  $Z_2$  (dots). Bottom panel: The hidden value  $Y_1$  (solid line), the kriging predictor of  $Y_1$  (dashed line), and the cokriging predictor of  $Y_1$  (dotted line).

```

Sigma_df <- expand.grid(s1 = df$s, comp1 = c("Y1", "Y2"), s2 = df$s, comp2 = c("Y1", "Y2")) %>%
  mutate(cov = c(Sigma))
Sigma_plot <- LinePlotTheme() +
  geom_tile(data=Sigma_df, aes(x=s2, y=s1, fill=cov)) +
  facet_grid(comp1 ~ comp2) +
  scale_fill_gradient2(low="white", high="black", mid="white") +
  coord_fixed() +
  ylab("s") + xlab("u") + scale_y_reverse() +
  theme(panel.margin = grid::unit(1, "lines"))
if(print_figs) ggsave(Sigma_plot,
                      filename = file.path(img_path,"Sigma.eps"),
                      width=8,height=7,family="Arial")
if(show_figs) print(Sigma_plot,width=16,height=7,family="Arial")

```

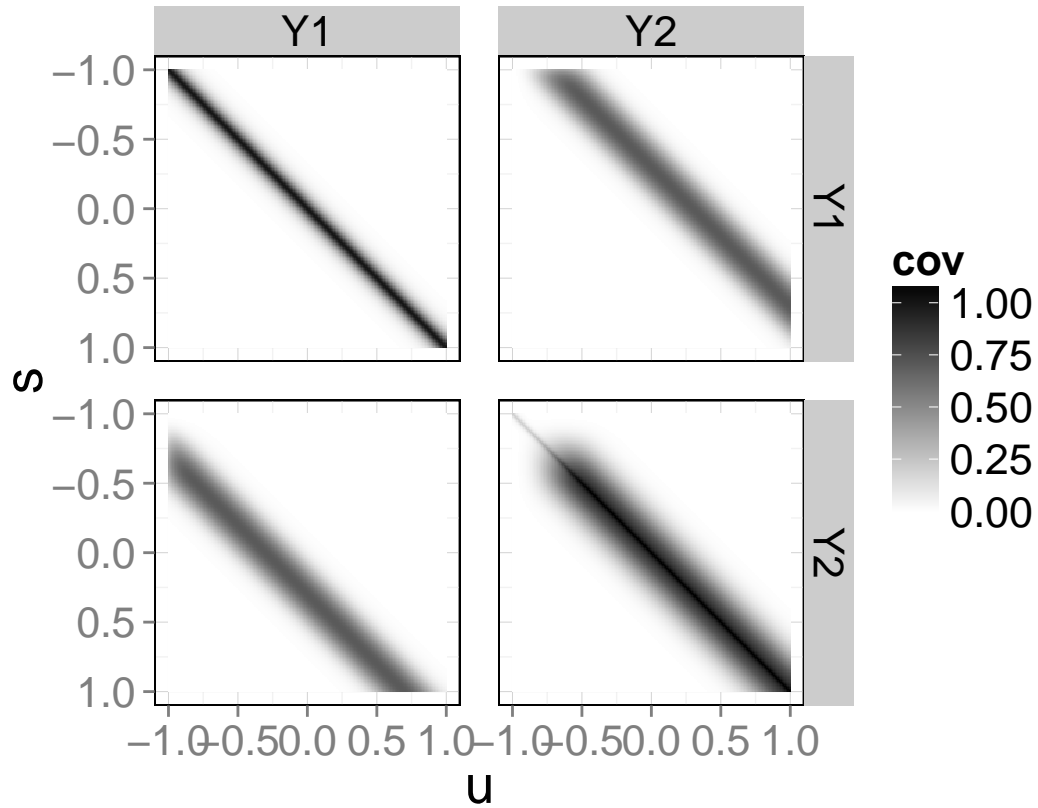


Figure 2: The covariance and cross-covariance matrix obtained using the function `makeSY`

```
g_all <- arrangeGrob(Sigma_plot,
  arrangeGrob(obs_plot, est_plotY1, ncol=1),
  ncol=2, widths=c(1,1))
if(print_figs) ggsave(g_all ,
  filename = file.path(img_path, "Fig1.eps"),
  width=16, height=7, family="Arial")
```

## References

- Cressie, N., & Zammit-Mangion, A. (2015). Multivariate spatial covariance models: A conditional approach. *Submitted*.
- Wackernagel, H. (1995). *Multivariate Geostatistics*. Berlin, DE: Springer.