Prediction and Validation

Denis Allard, **Liliane Bel**, Edith Gabriel, Thomas Opitz, Eric Parent

Workshop : An introduction to geostatistical analysis of spatio-temporal data with R

Montpellier, 12 July 2018

Spatio-temporal prediction

$Z(s,t)$ spatio-temporal Gaussian field, $s \in D$ location, $t \in (0,\infty)$ time.
Measurements : $z(s_i,t_j)$, $i=1,ns$, $j=1,T$
Best case : measurements are available for each time and each location,
generally there are missing values.

Spatio-temporal prediction:

- $\widehat{Z}(s_0,t_j)$
- $\widehat{Z}(s_i,T+h)$
- $\widehat{Z}(s_0,T+h)$

Models with trend

$$Z(s,t) = \mu(s,t) + \nu(s,t)$$

$$\mu(s,t) = \sum_{\ell=1}^{L} \gamma_\ell \mathcal{M}_\ell(s,t) + \sum_{i=1}^{m} \beta_i(s) f_i(t)$$

$\mathcal{M}_\ell(s,t)$ spatio-temporal covariates, γ_ℓ coefficients
$\{f_i(t)\}_i$ temporal function basis (singular value decomposition)
$\beta_i(s)$ spatially varying coefficients (universal kriging, depend on covariates), independent
$\nu(s,t)$ spatial field no temporal dependence, stationnary in space

Package *SpatioTemporal* (J. Lindstrom et al)

- basis selection $\{f_i(t)\}_i$
- missing data filling
- maximum likelihood estimation (profile, REML)
- prediction, variance
- cross validation

Models with trend

$$Z_t = X_t\beta + KY_t + e_t \qquad e_t \sim \mathcal{N}(0, \Sigma_e)$$

$$Y_t = GY_{t-1} + \eta_t \qquad \eta_t \sim \mathcal{N}(0, \Sigma_\eta)$$

$Y_0 \sim \mathcal{N}(m_0, C_0)$
Z_t : $ns \times 1$ measurements at stations
Y_t : $p \times 1$ latent process
X_t : $L \times ns$ covariates
e_t : random field, time independent, spatial covariance

Package *Stem* (M. Cameletti)

- parameter estimation by EM
- prediction
- variance error estimation by bootstrap
- example PM10

Spatio-temporal kriging

Z spatio-temporal covariance : $C((s_1, t_1), (s_2, t_2))$
Predict $Z(s_0, t_0)$ from Z with a linear combination
Minimizing the prediction variance leads to the simple kriging predictor

$$Z^*(s_0, t_0) = c(s_0, t_0)\Sigma^{-1}\mathrm{Z}$$

with

- $\mathrm{Z} = (Z(s_j, t_i))$, $i = 1, \ldots, ns$, $j = 1, \ldots, T$,
- $\Sigma = \mathbb{V}(\mathrm{Z})$
- $c(s_0, t_0) = \mathrm{Cov}(Z(s_0, t_0), \mathrm{Z})$

Kriging variance

$$\begin{aligned} \sigma^*(s_0, t_0) &= \mathbb{V}(Z^*(s_0, t_0) - Z(s_0, t_0))^{\frac{1}{2}} \\ &= (\mathbb{V}(Z(s_0, t_0)) - c(s_0, t_0)\Sigma^{-1}c(s_0, t_0))^{\frac{1}{2}} \end{aligned}$$

Computation complexity

If there are massive data, inverting Σ is computationally costly, may be unfeasable

Hints

- sliding neighborood
  - If there are not too much sites to predict
  - lead to discontinuities in the predicted map
  - the neighborhood has to be selected according to the parameters of the covariance model
- tapering (cf. covariance estimation)
  - product of the covariance function with a function with compact support in order to obtain a sparse covariance matrix
  - works well for dense dataset
  - boundary effects
- other possibilities : low rank models, spde

Validation

External Validation (EV)

- Split the dataset in two parts, training set and validation set

Cross Validation (CV)

- Leave-one-out cross validation
- K-fold CV (less costly)

For EV and K-fold CV, one can imagine subsets adapted to the spatio-temporal framework

- spatial : the same validation set of stations for all times
- temporal : all the stations for a subset of times

Prediction scores

- Mean square error and Normalized mean square error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{Z}_i^* - Z_i)^2 \text{ and } \mathrm{NMSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathrm{Z}_i^* - Z_i}{\sigma_i^*}\right)^2$$

- Logarithmic score

$$\mathrm{LogS} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\log(2\pi\sigma_i^{2*}) + \frac{1}{2}\left(\frac{\mathrm{Z}_i^* - Z_i}{\sigma_i^*}\right)^2\right)$$

- Continued Ranked Probability Score

$$\frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty}(F_i(y) - \mathbf{1}_{Z_i^* \leq y})dy$$

with $F_i(y) = \mathbb{P}(Z_i \leq y | Z_j, j \neq i)$

R implementation

- CompRandFld

  `Kri(...)`

  enables using tapering covariance
  non conditional simulations via `RFism`

- gstat

  `krigeST(...)`

  neighborhood may be defined by number max of neighbors,
  unique in the standard case
  no simulations

- RandomFields

  `RFinterpolate(...)`

  neighborhood defined by number max of neighbors
  conditional and non-conditional simulations via `RFismulate`

CompRandFld

```
Kri(loc,time,coordx,coordt,corrmodel, param, data)
```

with

- `data` : input data, matrix format
- `loc` : locations to predict coordinates, matrix format
- `time` : times to predict
- `coordx` : data coordinates
- `coordt` : data times
- `corrmodel` : correlation model
- `param` : correlation parameters

Returns a matrix of size $nrow(coordx) \times length(coordt)$.

gstat

```
krigeST(formula, data, newdata, modelList, beta,
        nmax = Inf, computeVar = FALSE)
```

with

- `formula` : for modelling the trend with possible covariates. ex `PM10~1` for ordinary kriging
- `data` : data, ST format
- `newdata` : locations to predict coordinates, ST format
- `modelList` : variogram model, vgmST format
- `beta` : known mean for simple kriging
- `nmax` : number max of neighbors for kriging with sliding neighborhood. Unique neighborhood standard case.
- `computeVar` : kriging variance computation
- other options useless

Returns an object class ST with the predicted values

RandomField

```
RFinterpolate(model, x, y = NULL, z = NULL, T = NULL,
grid=NULL, distances, dim, data, given=NULL,
err.model, ignore.trend = FALSE, ...)
```

with

- `model` covariance model with parameters
- `data` input data, array format (x_i, y_i, z_i, t_i, regular in t_i) or RFsp
- `x`, `y`, `z` locations to predict coordinates, regular in t
- `grid` locations to predict on a grid

Returns a vector of length $nx \times ny \times nt$