ifgi
Institute for Geoinformatics
University of Münster

Chapter 1

Different concepts of spatio-temporal kriging

Seminar *Analysis of Spatio-Temporal Data*, 2013-12-11

Benedikt Gräler
http://ifgi.de/graeler
Institute for Geoinformatics
University of Muenster

Proximity

Typically, we assume spatially close processes are stronger related than processes far apart (Tobler 's law).

We extend this assumption from the spatial S into the spatio-temporal space $S \times T$. Thus, we assume today's temperature is stronger related to tomorrow's temperature than to the temperatures next week.

The variogram

We call our spatial process $Z(s)$ and assume isotropy of the process, then the (semi)variogram is defined as

$$\gamma(h) = \mathbb{E}\big(Z(s) - Z(s+h)\big)^2$$

Extending to a twoplace function for spatio-temporal fields $Z(s,t)$:

$$\gamma(h,u) = \mathbb{E}\big(Z(s,t) - Z(s+h,t+u)\big)^2$$

at any location (s,t).

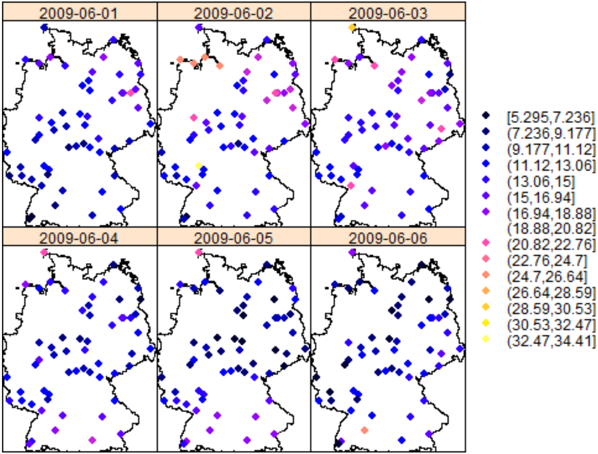
Asymmetry

Time has the special property of *one directional flow*.

Some processes will show a different pattern forward than backward in time. Imagine a sudden exposure of some toxic. Its concentration will suddenly rise but the decay will proceed slowly. Thus, the probability of observing a low value followed by a high one is higher than the probability of observing a low one after a high one.

Scenario

We have a set of spatially spread time series of daily measurements and are asked to produce a map of means for the provided time frame. The data is provided by the EEA, the presentation is composed along the lines of [2].



separate daily variograms

Every time slice is treated independently from the others.

This way, all the spatio-temporal data is used but no temporal interactions are incorporated. Predictions based on daily estimated variograms lead to a set of maps, the average per grid cell will provide the sought map of temporal means.

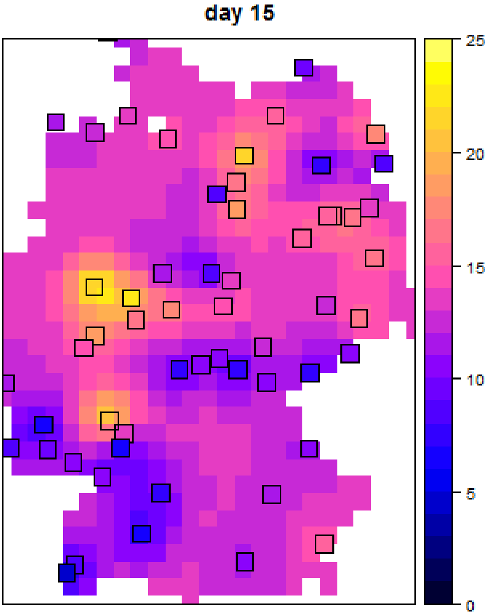
separate daily variograms in R

ifgi
Institute for Geoinformatics
University of Münster

Looping over all days

```
# day <- 0
# day <- day+1
tmpData <- ger_june[!is.na(ger_june[,day,"PM10"]@data),day]
vgmEmp <- variogram(PM10~1, tmpData, cutoff=150e3)
vgmModel <- fit.variogram(vgmEmp,vgm(4,"Sph",100e3,2))
plot(vgmEmp,vgmModel)

preds <- krige(PM10~1, tmpData, ger_gridded,vgmModel)


cols <- bpy.colors(25)[round(tmpData@data$PM10,0)]
spplot(preds,"var1.pred",col.regions=bpy.colors(25), at=0:25,
       sp.layout=list("sp.points",tmpData["PM10"], fill=cols,
                      pch=22, col="black", cex=2, lwd=1),
       main=paste("day",day))
```

kriged map for day 15 - separate days



daily evolving variograms

The estimated variogram does not only depend on the fit for the day but on the fits of the preceding days. For a given weight λ we set:

$$
\begin{aligned}
range = & \quad \lambda \cdot range_{curr} + (1-\lambda) \cdot range_{prev} \\
nugget = & \left(\lambda \cdot \frac{nugget_{curr}}{sill_{curr}} + (1-\lambda) \cdot \frac{nugget_{prev}}{sill_{prev}} \right) \cdot sill_{curr} \\
partial\ sill = & \quad sill_{curr} - nugget
\end{aligned}
$$

The kriging is performed time slice wise and the average over all maps generates the desired map. This method utilizes to some degree the temporal information and will add a "historical base" to days with a prevailing white noise process.

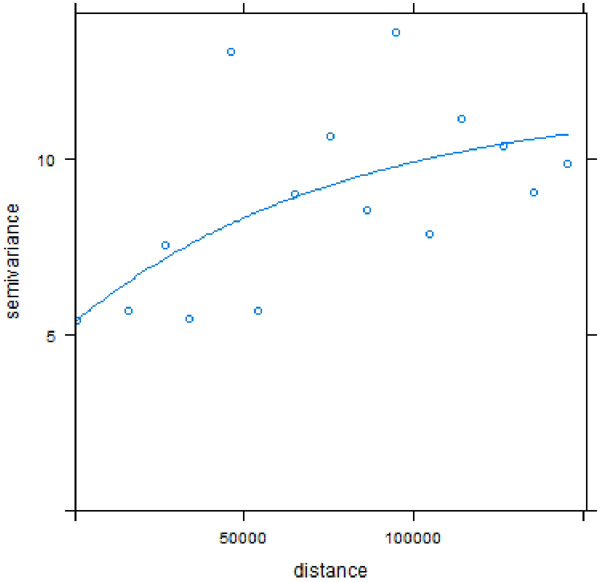
pooled and mean variogram

Instead of using a different variogram for each day, we assume a constant model and use one variogram relying on all data treating each day as a copy of the same process (*pooled variogram*) or averaging the variograms (*mean variogram*). The one estimated variogram is then used for each day. This way, all information is used but again no temporal interactions are incorporated.

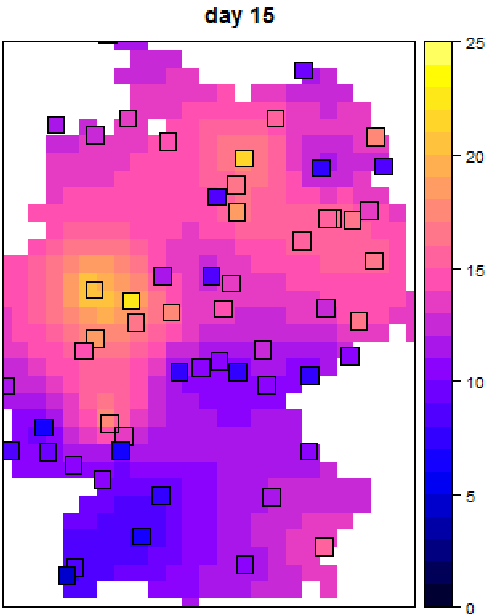
pooled variogram in R

```
lst = lapply(1:30, function(i) {
  x = ger_june[, i,"PM10"]
  x$ti = i
  x
})
pooledData = do.call(rbind, lst)

vgmPooledEmp = variogram(PM10 ~ ti,
  pooledData[!is.na(pooledData$PM10), ], dX=0, cutoff=150e3)
poolFit <- fit.variogram(vgmPooledEmp, vgm(psill=8,
  model="Exp", range=100e3, nugget=5))
plot(vgmPooledEmp,poolFit)

poolPreds <- krige(PM10~1,
  ger_june[!is.na(ger_june[,15,"PM10"]@data),15],
  ger_gridded, poolFit)

spplot(poolPreds,"var1.pred", col.regions=bpy.colors(25),
  at=0:25, sp.layout=list("sp.points",
    ger_june[!is.na(ger_june[,15,"PM10"]@data), 15],
    fill=cols, pch=22, col="black", cex=2),
  main=paste("day",15))
```

pooled variogram



kriged map for day 15 - pooled variogram



metric kriging

The *metric kriging* follows the natural idea of extending the 2-dimensional geographic space into a 3-dimensional spatio-temporal one. In order to achieve an isotropic space, the temporal domain has to be rescaled to match the spatial one (spatio-temporal anisotropy correction κ).

All spatial, temporal and sptio-temporal distances are treated equally resulting in a joint covariance model C_j:

$$C_m(h,u) = C_j\left(\sqrt{h^2 + (\kappa \cdot u)^2}\right)$$

The variogram evaluates to

$$\gamma_m(h,u) = \gamma_j(\sqrt{h^2 + (\kappa \cdot u)^2})$$

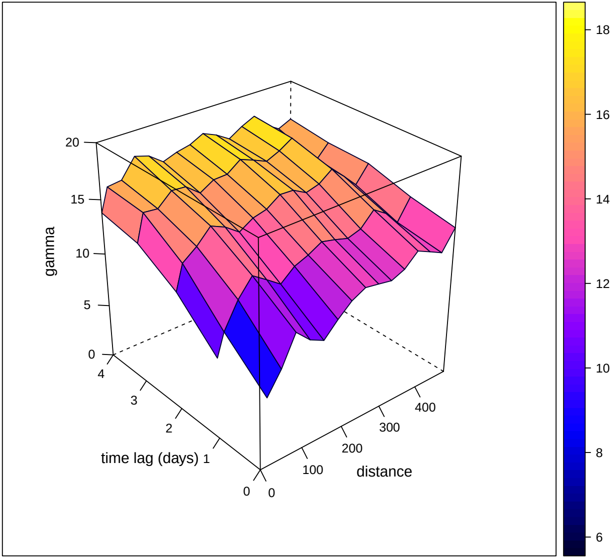
where γ_j is any known variogram including some nugget effect.

empirical spatio-temporal variogram surface

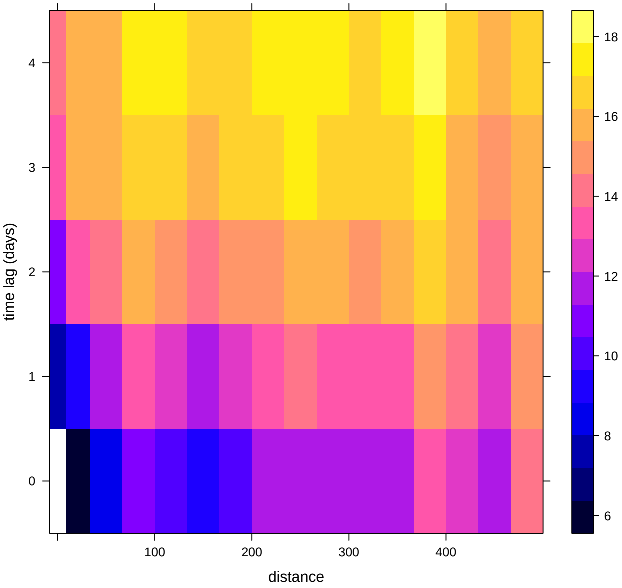
The idea is the same as in the spatial case: binning of locations according to their separating distance. In the spatio-temporal case, distances are pairs of spatial and temporal distance yielding a variogram surface, not a single line.

```
empVgm <- variogramST(PM10~1, ger_june, tlags=0:4,
                      cutoff=500e3)

# rescalig of distances
empVgm$dist <- empVgm$dist/1000
empVgm$spacelag <- empVgm$spacelag/1000

# wireframe:
plot(empVgm, wireframe=T, scales=list(arrows=F),
     col.regions=bpy.colors(),zlab=list(rot=90),zlim=c(0,20))

# levelplot:
plot(empVgm)
```

empirical spatio-temporal variogram surface - wireframe



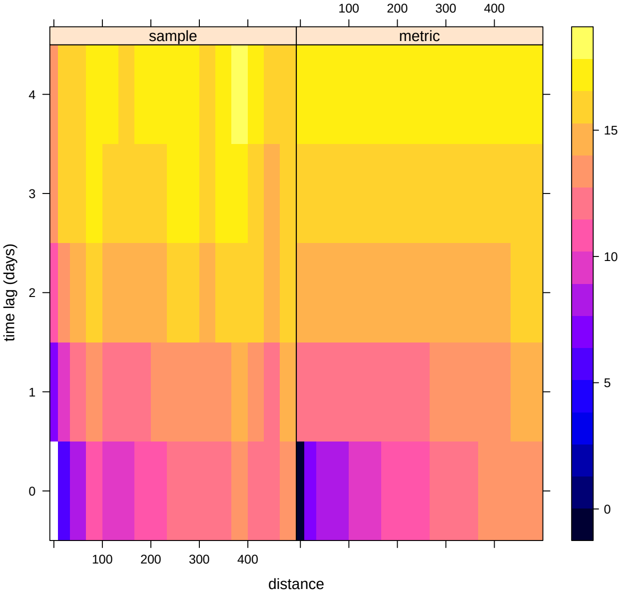
empirical spatio-temporal variogram surface - levelplot



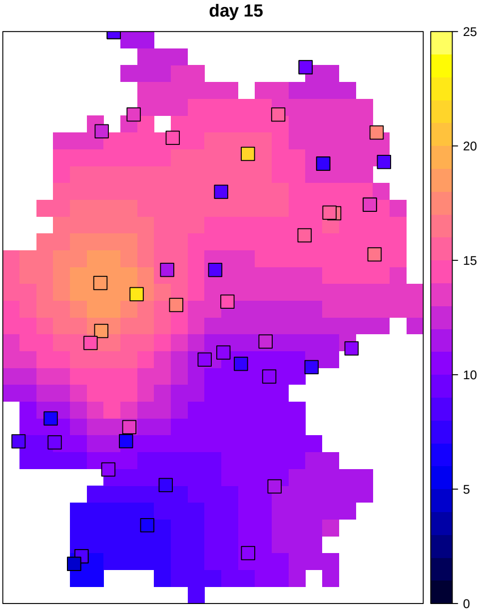
metric kriging in R

```
metricModel <- vgmST("metric",
                     joint=vgm(0.8,"Exp", 150, 0.2),
                     stAni=100)
metricFit <- fit.StVariogram(empVgm,metricModel,
                             lower=c(0,10,0,10))

attr(metricFit,"optim.output")$value
>  1.080641
plot(empVgm, metricFit)

predMetric <- krigeST(PM10~1, ger_june,
                      STF(ger_gridded,tgrd),
                      metricFit)
```

metric spatio-temporal variogram surface



kriged map for day 15 - metric model



separable covariance models

In space and under the assumptions of isotropy and stationarity, the covariance is a function $C(h)$ of the separating distance h between two locations. A spatio-temporal covariance function is thought of as a function of a spatial and a temporal distance $C(h,t)$.

A *separable covariance function* is assumed to fulfill $C_{sep}(h,u) = C_s(h)C_t(u)$. This is in general a rather strong simplification. Its variogram is given by

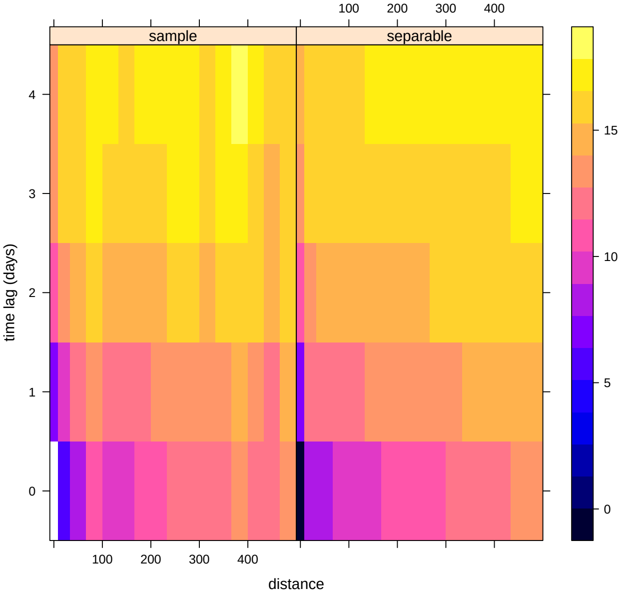
$$\gamma_{sep}(h,u) = \mathrm{nug}\cdot\mathbf{1}_{h>0,u>0} + \mathrm{sill}\cdot\left(\gamma_s(h)+\gamma_t(u)-\gamma_s(h)\gamma_t(u)\right)$$

where γ_s and γ_t are spatial and temporal variograms without nugget effect and a sill of 1. The overall nugget and sill parameters are denoted by "nug" and "sill" respectively.

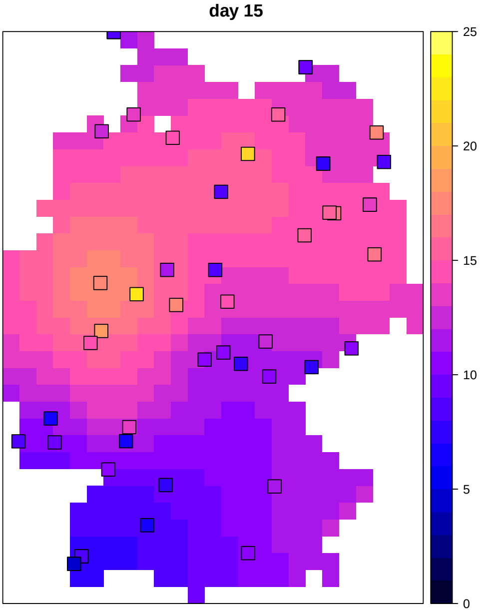
separable covariance model in R

```
sepModel <- vgmST("separable",
                  space=vgm(0.8,"Exp", 150, 0.2),
                  time =vgm(0.7,"Exp", 6, 0.3),
                  sill=18)
sepFit <-  fit.StVariogram(empVgm,sepModel,
                           lower=c(10,0,1,0,0))

attr(sepFit,"optim.output")$value
> 0.8001906
plot(empVgm, sepFit)

predSep <- krigeST(PM10~1, ger_june,
                   STF(ger_gridded,tgrd),
                   sepFit)
```

variogram surface of the product-sum model



kriged map for day 15 - seperable covariance model



product-sum covariance model

The *product sum covariance model* extends the simplifying assumption of the separable covariance model to ([1]):

$$C_{ps}(h,u) = k_1 C_s(h) + k_2 C_t(u) + k_3 C_s(h) C_t(u)$$

with $k_1 > 0$, $k_2 \geq 0$ and $k_3 \geq 0$ to fulfil the positive-definite condition. The corresponding variogram can be written as

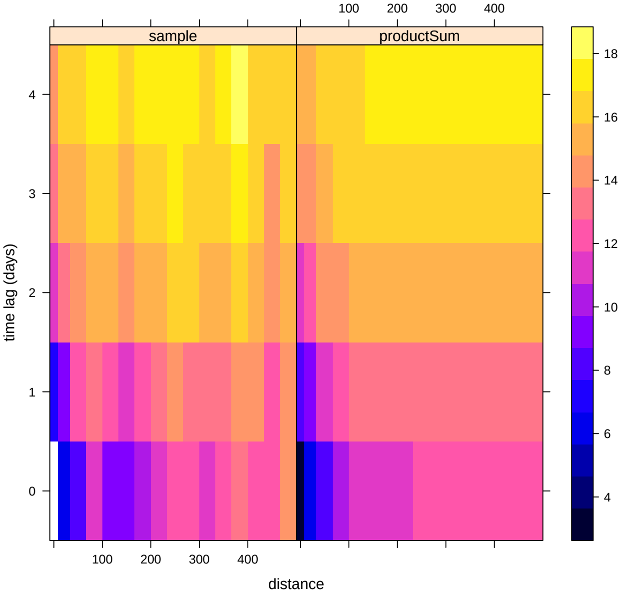
$$\gamma_{ps}(h,u) = \mathrm{nug} \cdot \mathbf{1}_{h>0,u>0} + \gamma_s(h) + \gamma_t(u) - k\gamma_s(h)\gamma_t(u)$$

where γ_s and γ_t are spatial and temporal variograms without nugget effect and in general different sills. The parameter k needs to fulfil $0 < k \leq 1/(\max(\mathrm{sill}_s, \mathrm{sill}_t))$ to let γ_{ps} be a valid model. The overall nugget is denoted by "nug".

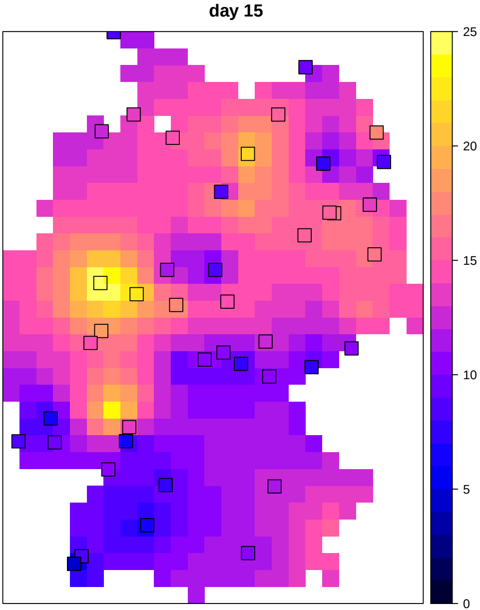
product-sum covariance model in R

```
empVgm$dist <- empVgm$dist/10
empVgm$spacelag <- empVgm$spacelag/10

psModel <- vgmST("productSum",
                 space=vgm(11,"Exp", 2),
                 time =vgm(5,"Sph", 6),
                 sill=16, nugget=5)
psFit <-  fit.StVariogram(empVgm,psModel)
attr(psFit,"optim.output")$value
> 0.7789366
plot(empVgm, psFit)

predPs <- krigeST(PM10~1, ger_june,
                 STF(ger_gridded,tgrd),
                 psFit)
```

variogram of the product-sum model



kriged map for day 15 - product-sum covariance model



sum-metric covariance model

The *sum-metric covariance model* is given by:

$$C_{sm}(h,u) = C_s(h) + C_t(u) + C_j\left(\sqrt{h^2 + (\kappa \cdot u)^2}\right)$$

Originally, this model allows for spatial, temporal and joint nugget effects, a simplified version may allow only for a joint nugget. The non-simplified variogram is given by

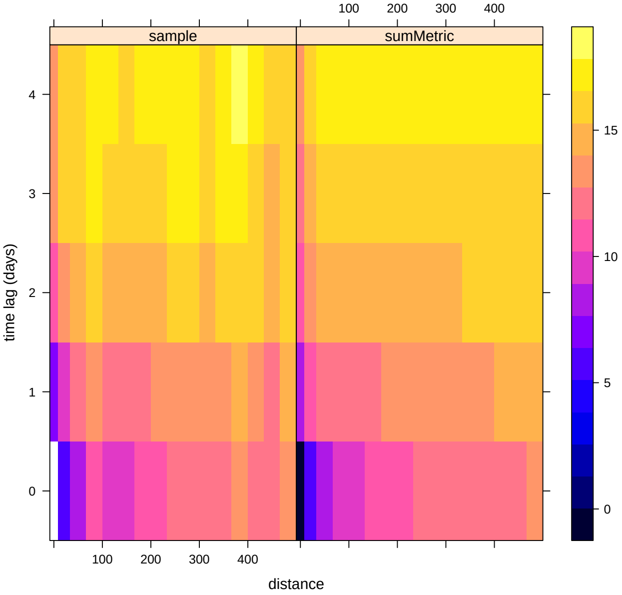
$$\gamma_{sm}(h,u) = \gamma_s(h) + \gamma_t(u) + \gamma_j(\sqrt{h^2 + (\kappa \cdot u)^2})$$

where γ_s, γ_t and γ_j are spatial, temporal and joint variograms with a separate nugget-effect.

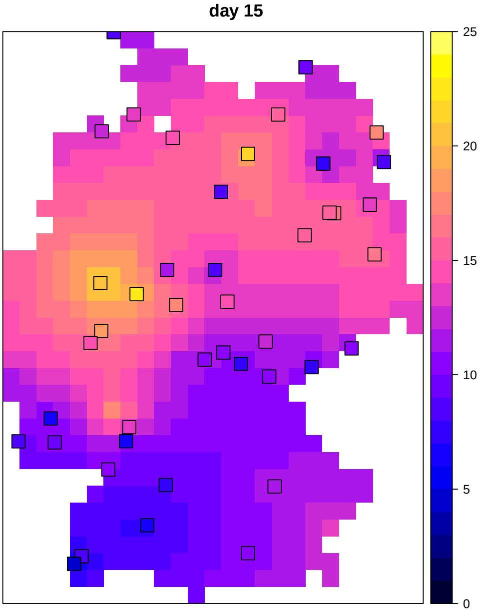
sum-metric covariance model in R

```
empVgm$dist <- empVgm$dist/10
empVgm$spacelag <- empVgm$spacelag/10

sumMetricModel <- vgmST("sumMetric",
                        space=vgm(5,"Exp",10,2),
                        time =vgm(5,"Exp", 6,2),
                        joint=vgm(5,"Exp",10,2),
                        stAni=10)
sumMetricFit <-  fit.StVariogram(empVgm,sumMetricModel,
                                 lower=c(0,1,0,0,1,0,0))
attr(sumMetricFit,"optim.output")$value
> 0.6754955

plot(empVgm, sumMetricFit)

predSumMetric <- krigeST(PM10~1, ger_june,
                         STF(ger_gridded,tgrd),
                         sumMetricFit)
```

variogram of the sum-metric model



kriged map for day 15 - product-sum covariance model



ifgi
Institute for Geoinformatics
University of Münster

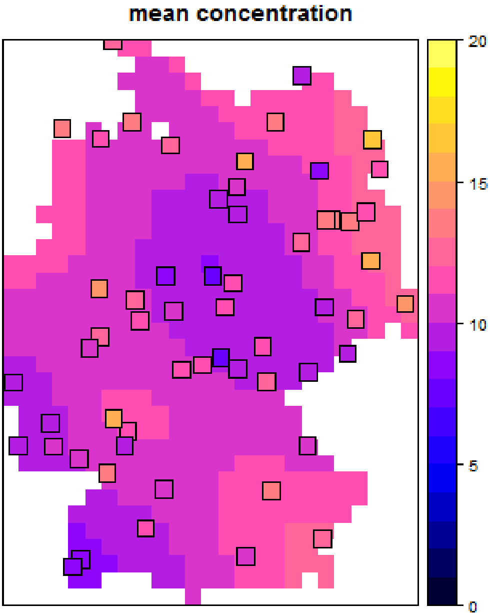
block kriging over time

In the above scenario and with the presented methods, it is hard to get an uncertainty estimate of the temporally averaged value. Block kriging, with blocks over time, is one way to get such estimates. However, one has to decide on a model beforehand. Here, we will use the metric model again.

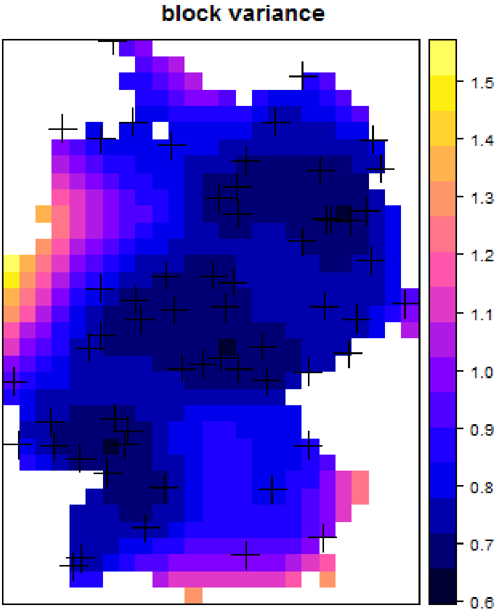
Block kriging does not provide estimates for single locations but for areas or volumes. It has the property of providing the correct kriging variance for the block estimate that is typically lower due to the larger area.

ifgi
Institute for Geoinformatics
University of Münster

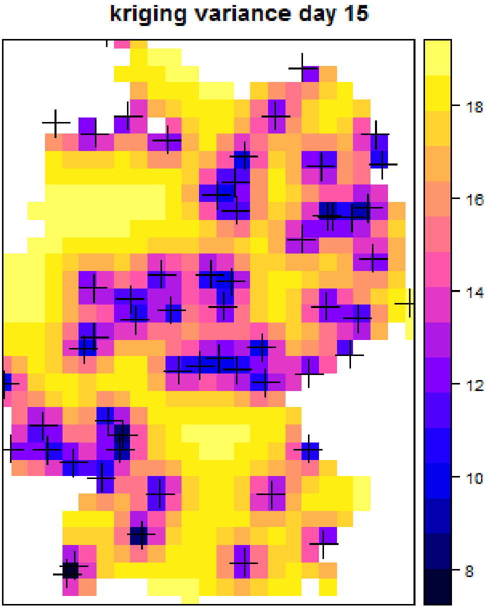
monthly mean concentration - block kriged



block kriging variance



kriging variance day 15



block kriging in R - metric workaround

ifgi
Institute for Geoinformatics
University of Münster

```
tmp_pred <- data.frame(cbind(ger_gridded@coords,15*tmpScale))
colnames(tmp_pred) <- c("x","y","t")
coordinates(tmp_pred) <- ~x+y+t

blockKrige <- krige(PM10~1,
                    air3d[as.vector(!is.na(air3d@data)),],
                    newdata=tmp_pred, model=model3d,
                    block=c(1,1,15*tmpScale))

ger_grid_time@sp@data <- blockKrige@data
```

local spatio-temporal kriging

Purely spatial kriging allows to select the n-nearest neighbours and use only these for prediction?

What does *nearest* mean in a spatio-temporal context?

The idea is to select the most *valuable* locations, i.e. the strongest correlated ones.

A first implementation is available in:

```
demo(localKrigeST)
```


ifgi
Institute for Geoinformatics
University of Münster

- De Iaco, S. (2001). Space–time analysis using a general product–sum model. Statistics Probability Letters, 52(1), pp. 21 - 28
- Gräler, B., L. E. Gerharz, & E. Pebesma (2012): Spatio-temporal analysis and interpolation of PM10 measurements in Europe. ETC/ACM Technical Paper 2011/10, January 2012.