Spatio-Temporal Interpolation using **gstat**

by Benedikt Gräler, Edzer Pebesma and Gerard Heuvelink

Mar 25, 2016

Abstract We present new spatio-temporal geostatistical modelling and interpolation capabilities of the R package **gstat**. Various spatio-temporal covariance models have been implemented, such as the separable, product-sum, metric and sum-metric models. In a real-world application we compare spatio-temporal interpolations using these models with a purely spatial kriging approach. The target variable of the application is the daily mean PM_{10} concentration measured at rural air quality monitoring stations across Germany in 2005. R code for variogram fitting and interpolation is presented in this paper to illustrate the workflow of spatio-temporal interpolation using **gstat**. We conclude that the system works properly and that the extension of **gstat** facilitates and eases spatio-temporal geostatistical modelling and prediction for R users.

1   Introduction

The collection and processing of spatio-temporal data is rapidly increasing due to technological advances and the societal need for analysis of variables that vary in space and time, such as weather and air quality variables, and crop yields. Analysis of spatial and temporal correlations is useful in itself to get insight into the character and causes of variability, but they are also important to predict values at points from neighbouring observations. Spatio-temporal interpolation can potentially provide more accurate predictions than spatial interpolation because observations taken at other times can be included. In addition, spatio-temporal interpolation allows predictions to be made at single locations or entire fields in between and beyond observation times. However, adding the temporal domain implies that variability in space and time must be modelled, which is more complicated than modelling purely spatial or purely temporal variability. The spatial, temporal and spatio-temporal dependence structures, for instance represented as variograms, do not necessarily coincide with each other in terms of their parameters nor in terms of their family. In the simplest case, a spatio-temporal anisotropy parameter might be enough to deal with the different dependence structures, but this poses strong assumptions on the process.

Interpolation of spatial random fields is a common task in geostatistics. Simple approaches like inverse distance weighted predictions or the well known kriging procedures have routinely been applied for many years. Nowadays, modern sensors allow to monitor different variables at an increasing temporal resolution producing rich spatio-temporal data sets. This calls as well for theory and methods to deal with these data sets to gain a better understanding of the observed spatio-temporal processes. While the theoretical aspects of spatio-temporal geostatistics show good progress [Cressie and Wikle, 2011], implementations lack behind. This hinders a wide application of spatio-temporal modelling, as typically extensive scripting and thorough understanding is necessary to build spatio-temporal models. Handling of spatio-temporal data in R is provided by the **spacetime** package [Pebesma, 2012]. In this paper, we present an extension of the **gstat** package [Pebesma, 2004] (version 1.1-3) that reuses the **spacetime** classes for the estimation of spatio-temporal covariance/variogram models and to perform spatio-temporal interpolation. Our implementation handles various types of spatio-temporal covariance structures and facilitates spatio-temporal interpolation. The notation of functions in **gstat** are extended in a way closely following the purely spatial design. This allows a researcher acquainted with **gstat** to readily use spatio-temporal tools. The use of the newly implemented functions is presented and illustrated by mapping spatio-temporal air-quality data. Another package that offers extensive spatio-temporal geostatistical functionality is **RandomFields** [Schlather et al., 2014]; further packages are mentioned in the SpatioTemporal CRAN task view.

The paper is organised as follows. The next section introduces the general interpolation routine and describes the different spatio-temporal covariance models, followed by a section introducing the German rural background data set for 2005 and performing the parameter estimation (i.e. covariance model fitting). Cross-validation results are presented and discussed in the section thereafter. Conclusions are drawn in the closing section. R scripts reproducing this study are available from within the **gstat** package as demos. `stkrige` re-estimates the variogram models, `stkrige-prediction`

re-executes the prediction for a time series and a couple of stations, and `stkrige-crossvalidation` re-runs the entire leave-one-out cross-validation (note that the latter takes a few hours).

2 Spatio-temporal dependence modelling and kriging

In the following, we will assume a Gaussian spatio-temporal random field Z defined over a spatial domain $\mathcal{S}$ and temporal domain $\mathcal{T}$. Typically, a sample $\mathbf{z} = (z(s_1, t_1), \ldots, z(s_n, t_n))$ has been observed at a set of distinct spatio-temporal locations $(s_1, t_1), \ldots, (s_n, t_n) \in \mathcal{S} \times \mathcal{T} \subseteq \mathbb{R}^2 \times \mathbb{R}$ that may include repeated measurements at the same location or simultaneous measurements at multiple spatial locations. Often, one is interested in modelling Z from the sample $\mathbf{z}$ in order to predict at unobserved locations in space and time or simulate from the conditional distribution.

Across our domain of interest $\mathcal{S} \times \mathcal{T}$, we assume the random field Z to be stationary and spatially isotropic. Hence, the field can be characterised through a mean μ and a covariance function C_{st} where the spatio-temporal covariance only depends on the separating distances across space $h \in \mathbb{R}_{\geq 0}$ and time $u \in \mathbb{R}_{\geq 0}$. Note that extensions beyond this set-up can easily be derived as has been done for the pure spatial case using for instance universal kriging to overcome the stationarity of the mean. The general spatio-temporal covariance function is given by $C_{\mathrm{st}}(h, u) = \mathrm{Cov}\left(Z(s, t), Z(\tilde{s}, \tilde{t})\right)$ for a separating spatial distance h and temporal distance u and any pair of points $(s, t), (\tilde{s}, \tilde{t}) \in \mathcal{S} \times \mathcal{T}$ with $||s - \tilde{s}|| = h$ and $|t - \tilde{t}| = u$. In general, this covariance function is hard to estimate but a couple of models using simplifying assumptions will be presented in the following together with their spatio-temporal variograms $\gamma_{\mathrm{st}}(h, u) = C_{\mathrm{st}}(0, 0) - C_{\mathrm{st}}(h, u)$ and encoding in **gstat**. Given a valid covariance function, the covariance matrices used in the linear predictor are easily obtained and the same algebraic operations as in the well known spatial case yield predictions of the desired random field [Cressie and Wikle, 2011]. A major difference is, however, the computational complexity of the matrix inversion. Typically, observations are made at a rather high temporal frequency leading to a number of spatio-temporal locations that is too large for global kriging. Hence, interpolation based on a selected neighbourhood of a subset of all data points becomes beneficial. Additionally, this relaxes the assumption of stationarity, as smooth variations in the mean value across the domain can be respected. The related class of dynamic models also addresses the computational complexity resulting in a temporal Markov structure. Implementations can be found in **spTimer** by Bakar and Sahu [2015], **spBayes** by Finley et al. [2015], **spate** by Sigrist et al. [2015] or **INLA** by Lindgren and Rue [2015].

2.1 Covariance models

The covariance models implemented in **gstat** and presented in this paper are introduced in the following. Besides further extensions we focus on the basic classes of the *separable*, *product-sum*, *metric* and *sum-metric* spatio-temporal covariance functions. The building blocks (in the following denoted as `spatialVgm`, `temporalVgm` or `jointVgm`) of the spatio-temporal covariance functions are any of the purely spatial variogram models already available in **gstat**. Each one of the building blocks is created by a call of the function `gstat::vgm()`. Remaining arguments such as `sill` (the joint sill), `nug` (the joint nugget component) or `stAni` (the spatio-temporal anisotropy used in the `jointVgm`) are scalars and refer to parameters of the entire spatio-temporal covariance function:

a) The *separable covariance model* assumes that the spatio-temporal covariance function can be represented as the product of a spatial and temporal term:

$$C_{\mathrm{sep}}(h, u) = C_s(h) C_t(u)$$

Its variogram is given by (see Appendix for details):

$$\gamma_{\mathrm{sep}}(h, u) = \mathrm{sill} \cdot \left(\bar{\gamma}_s(h) + \bar{\gamma}_t(u) - \bar{\gamma}_s(h)\bar{\gamma}_t(u)\right)$$

where $\bar{\gamma}_s$ and $\bar{\gamma}_t$ are standardised spatial and temporal variograms with separate nugget effects and (joint) sill of 1. The overall sill parameter is denoted by "sill".

The R package **gstat** encodes this model as:

`vgmST("separable", space = spatialVgm, time = temporalVgm, sill = sill)`

The separable model has a strong computational advantage in the setting where each spatial location has an observation at each temporal instance (a `"STFDF"` without `NA`s, Pebesma [2012]). In these cases, the covariance matrix (and its inverse) can be composed using the Kronecker-product of the purely spatial and purely temporal covariance matrices (respectively their inverse).

b) The above model extends to the *product-sum covariance model* that we give here in a slightly different notation as De Cesare et al. [2001] and De Iaco et al. [2001] by

$$C_{\mathrm{ps}}(h, u) = k C_s(h) C_t(u) + C_s(h) + C_t(u)$$

with $k > 0$.

The corresponding variogram can be written as

$$\gamma_{\text{ps}}(h, u) = (k \cdot \text{sill}_{\text{t}} + 1) \, \gamma_{\text{s}}(h) + (k \cdot \text{sill}_{\text{s}} + 1) \, \gamma_{\text{t}}(u) - k\gamma_{\text{s}}(h)\gamma_{\text{t}}(u)$$

where γ_{s} and γ_{t} are spatial and temporal variograms (see Appendix for details). The parameter k needs to be positive and the following identity defines the overall sill (sill_{st}) of the model in terms of the model's spatial and temporal sills:

$$\text{sill}_{\text{st}} = k \cdot \text{sill}_{\text{s}} \cdot \text{sill}_{\text{t}} + \text{sill}_{\text{s}} + \text{sill}_{\text{t}}$$

The above equation can also be used to estimate k based on the three sill values. An alternative formulation of the product-sum variogram can be found in De Iaco et al. [2001].

The **gstat** definition of this model reads:

```
vgmST("productSum", space = spatialVgm, time = temporalVgm, k = k)
```

c) Assuming identical spatial and temporal covariance functions except for spatio-temporal anisotropy, allows to use a spatio-temporal *metric covariance model* where, after matching space and time by an anisotropy correction κ (`stAni`), the spatial, temporal and spatio-temporal distances are treated equally resulting in a single covariance model C_{joint}:

$$C_{\text{m}}(h, u) = C_{\text{joint}} \left(\sqrt{h^2 + (\kappa \cdot u)^2} \right)$$

The variogram evaluates to

$$\gamma_{\text{m}}(h, u) = \gamma_{\text{joint}} \left(\sqrt{h^2 + (\kappa \cdot u)^2} \right)$$

where γ_{joint} (`jointVgm`) is any known variogram that may include a nugget effect. The following line generates the model in **gstat**:

```
vgmST("metric", joint = jointVgm, stAni = stAni)
```

The spatio-temporal anisotropy parameter κ (`stAni`) is given as spatial unit per temporal unit. In many cases, this will be in m/second, as these are the base units in our implementation. All temporal distances are hence internally re-scaled to an equivalent spatial distance using `stAni` and treated as metric 3D-space.

d) A combination of spatial, temporal and a metric model including an anisotropy parameter κ is found in Bilonick [1988] and revisited by Snepvangers et al. [2003] as the *sum-metric covariance model*:

$$C_{\text{sm}}(h, u) = C_{\text{s}}(h) + C_{\text{t}}(u) + C_{\text{joint}} \left(\sqrt{h^2 + (\kappa \cdot u)^2} \right)$$

This model allows for spatial, temporal and joint nugget effects. Thus, the variogram is given by

$$\gamma_{\text{sm}}(h, u) = \gamma_{\text{s}}(h) + \gamma_{\text{t}}(u) + \gamma_{\text{joint}} \left(\sqrt{h^2 + (\kappa \cdot u)^2} \right)$$

where γ_{s}, γ_{t} and γ_{joint} are spatial, temporal and joint variograms with separate nugget-effects. This model can be defined in **gstat** through:

```
vgmST("sumMetric", space = spatialVgm, time = temporalVgm, joint = jointVgm,
      stAni = stAni)
```

e) A simplified version of the above model is to restrict the spatial, temporal and joint variograms to nugget free models. Additionally, a single spatio-temporal nugget is introduced and the variogram takes the representation:

$$\gamma_{\text{ssm}}(h, u) = \text{nug} \cdot \mathbf{1}_{h>0 \vee u>0} + \gamma_{\text{s}}(h) + \gamma_{\text{t}}(u) + \gamma_{\text{joint}} \left(\sqrt{h^2 + (\kappa \cdot u)^2} \right)$$

The *simple sum-metric covariance model* can be obtained by:

```
vgmST("simpleSumMetric", space = spatialVgm, time = temporalVgm,
      joint = jointVgm, nugget = nug, stAni = stAni)
```

Note that the above mentioned spatial, temporal and joint components of the spatio-temporal covariance and variogram models need not necessarily exhibit the same structure. Taking for instance the product-sum and sum-metric models that both contain single temporal and spatial variogram models: the best fits of the respective spatio-temporal models might suggest different variogram families and parameters for the pure spatial and temporal ones. This is due to the target of finding the best overall variogram surface resulting in (potentially) different marginal models.

Table 1: List of implemented weighting schemes for variogram optimisation. Methods 3, 4, and 5 are kept for compatibility reasons with the purely spatial `fit.variogram` function. The following notation is used: N_j number of pairs, h_j mean spatial distance and u_j mean temporal distance for each bin j, γ the actual proposed variogram model and `stAni` a spatio-temporal anisotropy scaling.

fit.method	weights
0	no fitting
1 and 3	N_j
2 and 4	$N_j / \gamma\left(h_j, u_j\right)^2$
5	reserved for REML
6	1, no weighting
7	$N_j / \left(h_j^2 + \text{stAni}^2 \cdot u_j^2\right)$
8	N_j / h_j^2
9	N_j / u_j^2
10	$1 / \gamma\left(h_j, u_j\right)^2$
11	$1 / \left(h_j^2 + \text{stAni}^2 \cdot u_j^2\right)$
12	$1 / h_j^2$
13	$1 / u_j^2$

2.2 Parameter estimation

Fitting routines for the above variogram models are implemented in **gstat** through the function `fit.StVariogram()`, which calls `optim()` from the R core package **stats**. Additional parameters to improve the numerical optimisation can be provided to `fit.StVariogram()` and will be passed on to `optim()` (using R's three-dots mechanism). As some of the parameters are limited to certain ranges (nuggets need to be non-negative, ranges must be positive), it is advised to use an optimisation routine that allows to impose limits on the search space (i.e. `L-BFGS-B`) and provide sensible limits via `lower` and `upper`. By default, the method `L-BFGS-B` is called and the smallest lower and largest upper bounds supported by the model are given. The estimation of the spatio-temporal variogram models relies on a sample variogram empirically derived from the data. In contrast to the spatial variogram line, the spatio-temporal variogram is represented by a surface for lag classes composed of a spatial and temporal separation distance. Different from the spatial case, a spatio-temporal sample variogram contains lag-classes of zero spatial separation describing pure temporal dependencies. Without duplicate observations, no estimates can be made for the lag-class with both zero spatial and zero temporal separation. The sample variogram is calculated through the function `variogram()` that dispatches the call for spatio-temporal data objects (of class `"STFDF"`, `"STSDF"`, or `"STIDF"`) from **spacetime**. For a visual judgement of the fit between sample and fitted variograms the `plot()` function can be called to show the variogram surfaces next to each other as coloured level plots. Alternatively, a wireframe plot is obtained by setting the parameter `wireframe = TRUE` (Figure 3). A further option is to plot the differences between the sample and model variogram surfaces by setting `diff = TRUE`, see Figure 4. Additionally to visual comparison, `fit.StVariogram()` provides the output of `optim` as attribute `optim.out` of the returned S3-class `"StVariogram"`. This attribute includes valuable information to diagnose the success of the `optim` routine. It contains for instance the convergence code (`$convergence`) or message (`$message`) and the optimised value (`$value`), which is the mean of the (weighted) squared deviations between sample and fitted variogram surface. Furthermore, it is advised to check the estimated parameters against the parameter boundaries and starting values. Additionally, starting values might also influence the success and result of the optimisation, as local optima may occur due to the interdependence of the parameters. Alternatively, the user might want to start a grid search in order to better asses the sensitivity of the estimates.

The fitting approach is identical for all covariance models. However, with the flexibility of the model also the number of parameters increases, making a numerical estimation at times cumbersome. Starting values can in most cases be read from the sample variogram. Parameters of the spatial and temporal variograms can be assessed from the spatio-temporal surface fixing the counterpart at 0. The overall spatio-temporal sill including the nugget can be deducted from the plateau that a nicely behaving sample variogram reaches for 'large' spatial and temporal distances. An important issue is the potentially different orders of magnitude of the parameters. It is at times advisable to rescale spatial and temporal distances to ranges similar to the ones of sills and nuggets using the parameter `parscale`. `parscale` needs to be provided via `control = list(parscale=...)` and holds a vector of the same length as the number of parameters to be optimised (see the documentation of `optim` for further details).

Currently, the implemented fitting routines are based on the (weighted) mean squared difference between model and sample variogram surfaces. By default, all values are associated the same weight
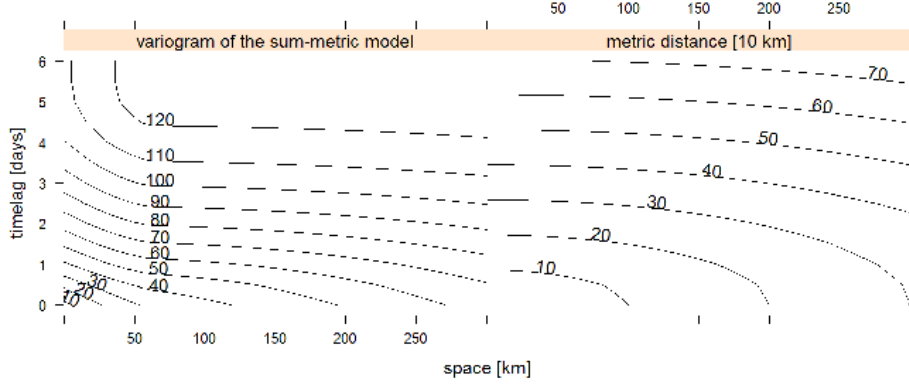
Figure 1: A contourplot showing how the spatio-temporal sum-metric variogram model (as estimated in the application below) and a metric distance relate to each other. Distances are rescaled by 1/5 for easy plotting.

(`fit.method = 6`), but other options are available that allow for different weighting schemes based on the number of pairs, spatial, temporal and spatio-temporal distances or the variogram's value. Table 1 lists all currently implemented options. Depending on the target neighbourhood size of the desired interpolation, it might be beneficial to narrow down the spatial and temporal distances and to introduce a cutoff. This ensures that the model is fitted to the differences over space and time actually used in the interpolation, and reduces the risk of overfitting the variogram model to large distances not used for prediction. Please note that methods 2 and 10 (Table 1) involve weights based on the fitted variogram that might lead to bad convergence properties of the parameter estimates. Furthermore, the scalar `stAni` in methods 7 and 11 will either be the actual fitted spatio-temporal anisotropy if it is included in the model or a fixed value that has to be passed as `stAni` by the user to `fit.StVariogram`. The latter is advised, as the former might lead to bad convergence properties as in the case of weights based on the fitted variogram mentioned above. As the estimation of an anisotropy scaling might be cumbersome on a visual basis, we provide the function `estiStAni` that provides estimates based on the empirical spatio-temporal variogram. Four heuristics are available based on (i) rescaling a linear model (`linear`), (ii) estimating equal ranges (`range`), (iii) rescaling a pure spatial variogram (`vgm`) or (iv) estimating a complete spatio-temporal metric variogram model and returning its spatio-temporal anisotropy parameter (`metric`). The choice of the weighting scheme will influence the selected model and different weightings might be further assessed in a cross-validation of the selected model. To increase numerical stability, it is advisable to use weights that do not change with the current model fit.

2.3 Kriging

Standard kriging (`krigeST`) and trans Gaussian kriging (`krigeSTTg`) have been implemented. As spatio-temporal kriging based on the complete data set might be too computationally expensive, local kriging is an attractive alternative. This poses the question of how to select the "nearest" neighbours from the spatio-temporal space $\mathcal{S} \times \mathcal{T}$. A natural choice would be to select the spatio-temporal locations exhibiting the strongest correlation to the unobserved location. Depending on the spatio-temporal covariance model, the relation between spatial and temporal distance in determining the strength of correlation will vary. As a proxy, we use a spatio-temporal anisotropy parameter that relates spatial and temporal distances in the same manner as in the metric covariance models. The k-nearest neighbours within this metric spatio-temporal space $\mathcal{S} \times \mathcal{T}$ are selected using the R package **FNN** [Beygelzimer et al., 2013]. The interpolation performs iteratively for each spatio-temporal prediction location with a local subset of the data set. Without neighbourhood selection, kriging uses all data.

As the metric induced by the spatial and rescaled temporal distances are only proxies to the strength of correlation between locations (see Figure 1), we provide an option to search a larger metric neighbourhood. Within this larger neighbourhood, the covariance function is evaluated for all spatio-temporal locations and the neighbouring locations with the largest covariance values are then selected for prediction. However, this approach might still suffer from a clustering of locations and alternatives such as a staged search (find spatial neighbours first and select a set of temporal instances for each spatial neighbour) or an octant search (select neighbours per spatial quadrant from preceding and following time stamps separately) could be considered. However, these alternatives are not yet available in **gstat**.
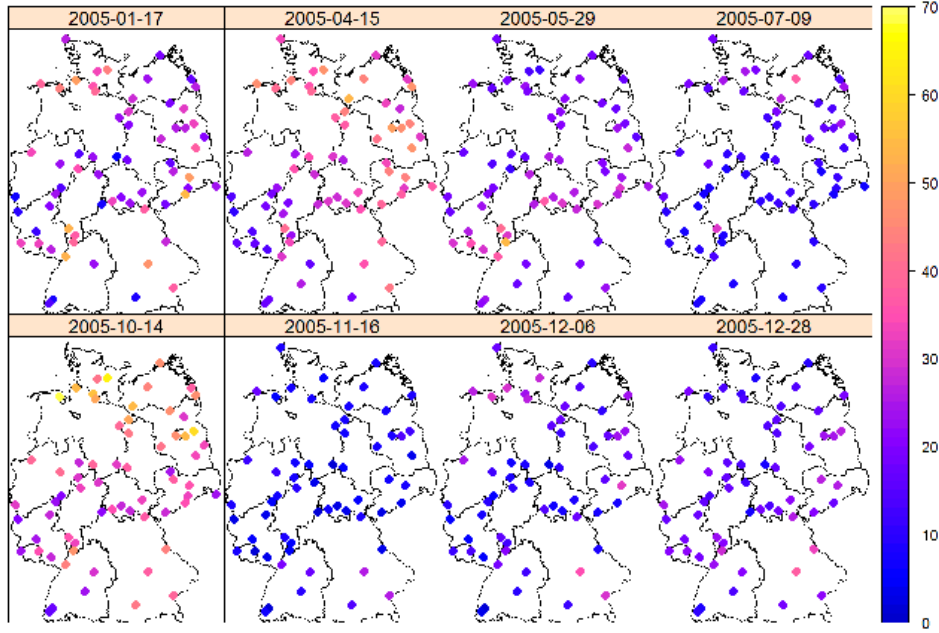
Figure 2: Daily mean PM_{10} concentration [$\mu g/m^3$] at 8 randomly selected days in 2005.

3 Application and illustration

The data set used is taken from AirBase[1], the air quality data base for Europe provided by the European Environmental Agency (EEA). We focus on a single air quality indicator, particulate matter with a diameter less than 10 μm, measured at rural background stations for 2005 (PM_{10}). The data base contains data for many years. Besides rural, also urban areas are monitored and not only at background locations (e.g. traffic stations). However, these processes are considered to be of a different nature and should be treated separately. As a use case, we therefore limit our data set to the rural background stations in Germany. Figure 2 shows for 8 randomly chosen days daily mean values of PM_{10} concentrations for the entire monitoring network over Germany in 2005 with 69 rural background stations.

In order to fit a spatio-temporal model to the air quality data set, the empirical variogram surface is computed and used as input for the fitting routines of the different models. The empirical variogram is based on spatio-temporal bins that span regularly over space and time.

Regular measurements over time (i.e. hourly, daily) motivate regular binning intervals of the same temporal resolution. Nevertheless, flexible binning boundaries can be passed for spatial and temporal dimensions. This allows for instance to use smaller bins at small distances and larger ones for large distances. Temporal boundaries, instead of lags, are required when the sampling of the data is non-regular. In cases where regular temporal observations can be assumed, this is utilised in the sample variogram calculations and any two temporal consecutive observations are assumed to have the same temporal distance. Figure 3 shows the empirical variogram along with the proposed best fitting model of each spatio-temporal variogram family as perspective wireframe plots. In order to better identify structural shortcomings of the selected model, a difference plot (Figure 4) is a helpful visual diagnostic plot.

Beyond the selection of the spatio-temporal variogram family, each component of this model can be chosen from any implemented one-dimensional variogram. In Table 2 a selection of fitted models in terms of their residuals compared to the sample variogram surface are shown. The best fitting spatio-temporal model of each family is given as:

a) separable model (weighted MSE: 6.82):

	partial sill	model	range	nugget	sp.-temp. sill
space	0.86	Exp	558 km	0.14	124
time	1.00	Sph	5.6 days	0.00	

obtained via:

```
separableModel <- vgmST("separable",
```

---

[1] AirBase - The European air quality database

```
                              space = vgm(0.9, "Exp", 200, 0.1),
                              time = vgm(0.9, "Sph", 3.5, 0.1),
                              sill = 124)
    fit.StVariogram(empVgm, separableModel, fit.method = 7,
                    stAni = 117.3, method = "L-BFGS-B",
                    control = list(parscale = c(100, 1, 10, 1, 100)),
                    lower = c(10, 0, 0.1, 0, 0.1),
                    upper = c(2000, 1, 12, 1, 200))
```

b) product-sum model (weighted MSE: 6.91)

	partial sill	model	range	nugget	k
space	6.8	Exp	542 km	1.2	
time	8.7	Sph	5.5 days	0.0	1.61

obtained via

```
prodSumModel <- vgmST("productSum",
                      space = vgm(10, "Exp", 200, 1),
                      time = vgm(10, "Sph",   2, 1),
                      k=2)
    fit.StVariogram(empVgm, prodSumModel, fit.method = 7,
                    stAni = 117.3, method = "L-BFGS-B",
                    control = list(parscale = c(1, 10, 1, 1, 0.1, 1, 10)),
                    lower = rep(0.0001, 7))
```

c) metric model (weighted MSE: 10.05)

	partial sill	model	range	nugget	anisotropy
joint	123.4	$Mat_{\kappa=0.6}$	453 km	17.4	189 km/day

obtained via

```
metricModel <- vgmST("metric",
                     joint = vgm(60, "Mat", 150, 10, kappa = 0.6),
                     stAni = 60)
    fit.StVariogram(empVgm, metricModel, fit.method = 7,
                    stAni = 117.3, method = "L-BFGS-B",
                    control = list(parscale = c(10, 20, 5, 10)),
                    lower = c(80, 50, 5, 50),
                    upper = c(200, 1500, 60, 300))
```

d) sum-metric model (weighted MSE: 3.31)

	partial sill	model	range	nugget	anisotropy
space	16.4	Sph	67 km	0	
time	9.3	Exp	0.9 days	0	
joint	91.5	Sph	999 km	7.3	185 km/day

obtained via

```
sumMetricModel <- vgmST("sumMetric",
                        space = vgm(20, "Sph", 150, 1),
                        time = vgm(10, "Exp", 2, 0.5),
                        joint = vgm(80, "Sph", 1500, 2.5),
                        stAni = 120)
    fit.StVariogram(empVgm, sumMetricModel, fit.method = 7,
                    stAni = 117.3, method = "L-BFGS-B",
                    control = list(parscale = c(1, 100, 1, 1, 0.5, 1,
                                                  1, 100, 1, 100),
                                   maxit=10000),
                    lower = c(sill.s = 0,  range.s = 10, nugget.s = 0,
                              sill.t = 0, range.t = 0.1, nugget.t = 0,
                              sill.st = 0, range.st = 10, nugget.st = 0,
                              anis = 40),
                    upper = c(sill.s = 200, range.s = 1000, nugget.s = 20,
                              sill.t = 200, range.t = 75, nugget.t = 20,
                              sill.st= 200, range.st = 5000, nugget.st = 20,
                              anis = 500))
```

Table 2: Weighted MSE (`fit.method = 7, see Table 1`) for different spatio-temporal variogram families and different choices for the one-dimensional variogram components. Columns denote the spatial and temporal variogram choices. The metric model is only applicable if both domains use the same family.

model	joint	Exp+Exp	Exp+Sph	Sph+Exp	Sph+Sph	Mat$_{\kappa=0.6}$
separable	·	9.87	**6.82**	10.42	7.50	·
product-sum	·	10.09	**6.91**	10.64	7.59	·
metric	·	10.25	·	·	10.59	**10.05**
sum-metric	Exp	4.10	3.60	3.89	3.32	·
	Sph	3.74	3.73	**3.31**	3.36	·
simple sum-metric	Exp	4.10	3.60	3.94	3.32	·
	Sph	3.74	3.98	**3.31**	3.56	·

e) simple sum-metric model (weighted MSE: 3.31)

	partial sill	model	range	anisotropy	sp.-temp. nugget
space	16.4	Sph	67 km		
time	9.3	Exp	0.9 days		7.3
joint	91.5	Sph	999 km	185 km/day	

obtained via

```
simpleSumMetricModel <- vgmST("simpleSumMetric",
                        space=vgm(120, "Sph", 150),
                        time =vgm(120, "Exp", 10),
                        joint=vgm(120, "Sph", 150),
                        nugget = 10, stAni = 150)
fit.StVariogram(empVgm, simpleSumMetricModel, fit.method = 7,
            stAni = 117.3, method = "L-BFGS-B",
            control = list(parscale = c(1, 10, 1, 1, 1, 100, 1, 10))
            lower = c(sill.s = 0, range.s = 10,
                    sill.t = 0, range.t = 0.1,
                    sill.st= 0, range.st= 10,
                    nugget = 0, anis = 40),
            upper = c(sill.s = 200,  range.s = 500,
                    sill.t = 200,  range.t = 20,
                    sill.st= 200, range.st = 5000#,
                    nugget = 100, anis = 1000))
```

The variogram parameters are numerically optimised using the function `fit.StVariogram` and the `L-BFGS-B` routine of `optim`. The parameter `fit.method` that controls the weighing of the residuals between empirical and model surface of `fit.StVariogram` is set to `7` (the spatio-temporal analog to the commonly used spatial weighting). A full list of all weighting schemes is presented in Table 1. In our application, the residuals are multiplied by the number of pairs in the corresponding spatio-temporal bin divided by the metric distance: $N_j/(h_j^2 + \text{stAni}^2 \cdot u_j^2)$. The spatio-temporal anisotropy is estimated beforehand and fixed at 118 km/day. This weighting scheme puts higher confidence in lags filled with many pairs of spatio-temporal locations, but respects to some degree the need of an accurate model for short distances, as these short distances are the main source of information in the prediction step. Note, that different weighting schemes will in general result in different model parameters generating different interpolation values. Our selection is based on the assumption that well filled bins provide more reliable empirical variogram estimates and the fact that short distances are the most important ones for a local interpolation.

For comparison with classical approaches, we interpolate across Germany iteratively for each single day using all available data for variogram estimation. The purely spatial empirical variogram can directly be obtained from the empirical spatio-temporal variogram, by fixing the temporal lag at 0 separation. From the same set of variogram models as investigated for the spatio-temporal models, the exponential model (partial sill: 66.5, range: 224 km, nugget: 13.5) is the best suited based on the optimisation criterion. Alternatively, we could have fitted the spatial variogram for each day separately using observations from that day only. However, given the small number of observation stations, this produced unstable variograms for several days and we decided to use the single spatial variogram derived from all spatio-temporal locations treating time slices as uncorrelated copies of the spatial random field.
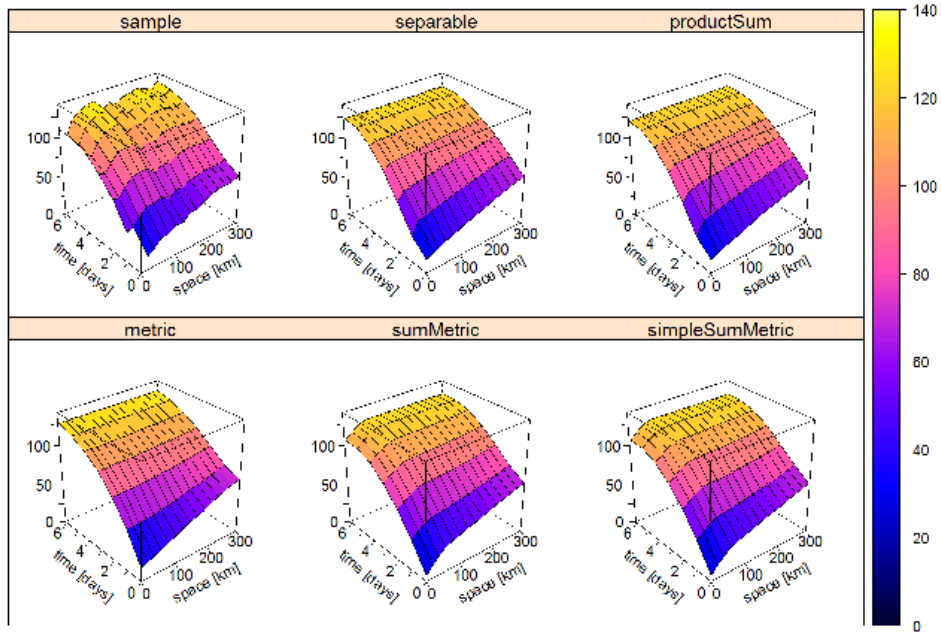
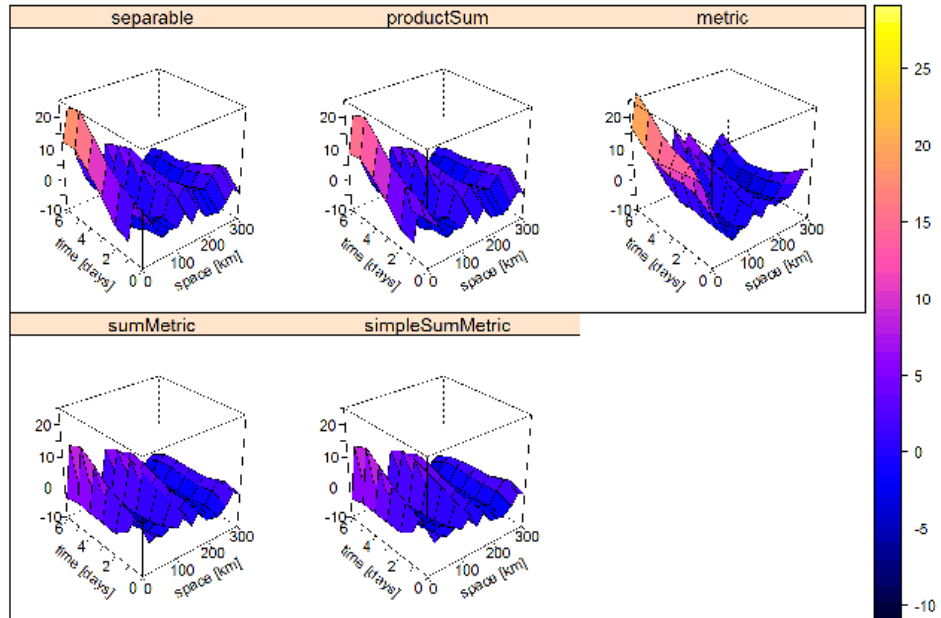**Figure 3:** Sample and the best fitting spatio-temporal variogram of each family.

Figure 4: Differences between the sample and the best fitting spatio-temporal variogram of each family.

Once the best fitting spatio-temporal variogram model is identified, the interpolation can be executed with the help of the function `krigeST`. We use the sum-metric model that obtained the smallest RMSE (compare Table 2) to produce a gridded prediction. The interpolation domain consists of daily values for a regular grid spanning over Germany in UTM projection. The cell size is 10 km × 10 km. Figure 5 shows the interpolated grid for the same days as Figure 2 alongside with all sampling locations. Additionally, maps depicting the differences from a leave-one-out cross-validation are presented in Figure 6. A time series view is presented in Figure 7 showing the observed and predicted time series at a single location along with its 95 % prediction intervals. An animation of the entire year of daily mean PM_{10} prediction maps can be viewed online.[2]

The interpolated maps are generated for a set of time stamps `tIDs` and a grid over Germany `DE_pred` by

```
krigeST(PM10~1, data = DE_RB_2005[ , tIDS],
        newdata = DE_pred, fitSumMetricModel, nmax = 50,
        stAni = fitMetricModel$stAni/24/3600)
```

To further compare the different approaches, a leave-one-out cross-validation was carried out. The spatio-temporal interpolations are done for the closest 50 and 10 neighbours assessing the impact of the neighbourhood size. Inspection of the ranges of the variograms in the temporal domain, suggests that any station more than at most 6 days apart does not meaningfully contribute. Furthermore, the local estimation allows the spatio-temporal random field to have a varying mean value over space and time. The purely spatial interpolation can be considered as the extreme temporally local case, where only observations from the same time instance are considered.

4   Results and discussion

In terms of added value of spatio-temporal kriging measured in cross-validation results, Table 3 shows hardly any benefit in the illustrative example. This effect can to a large degree already be explained from the spatio-temporal variograms: a temporal lag of one or a few days leads already to a large variability compared to spatial distances of few hundred kilometres, implying that the temporal correlation is too weak to considerably improve the overall prediction. Nevertheless, investigating a process with a higher temporal frequency will likely show a stronger correlation in the temporal domain. Looking into station-wise cross-validation statistics (not shown), the four
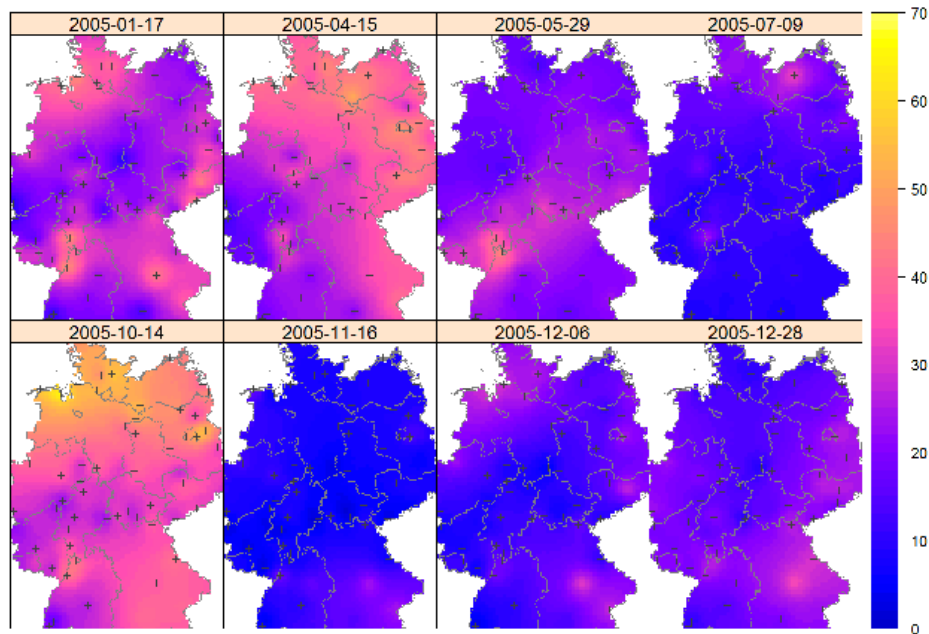
---

[2]https://gstat.r-forge.r-project.org/STpred.html

Figure 5: Spatio-temporal interpolation of daily mean PM_{10} concentrations using the sum-metric covariance model with the closest 50 neighbouring spatio-temporal locations. The crosses indicate sampling locations. The cell size of the grid in UTM projection is 10 km × 10 km.
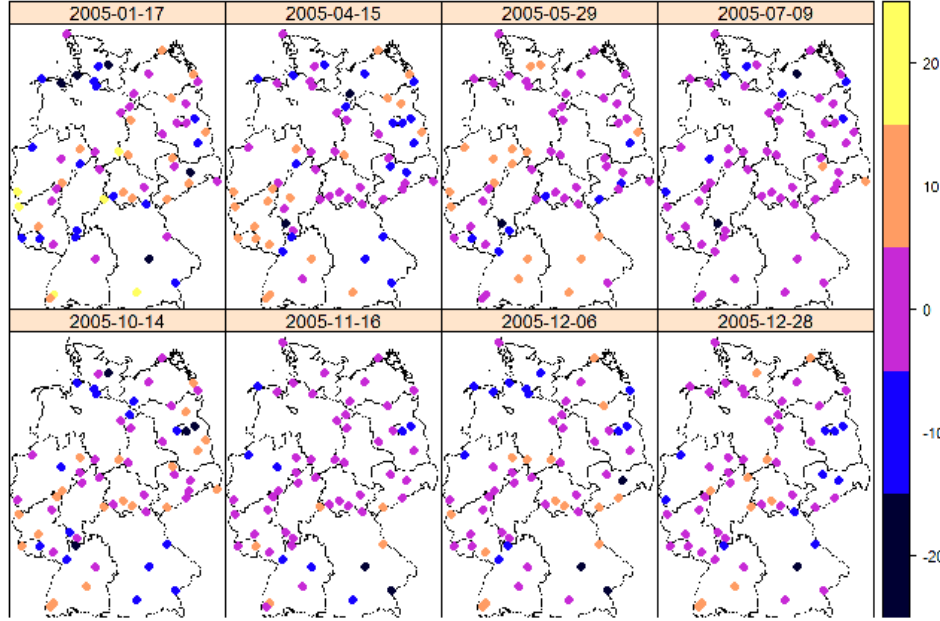
Figure 6: Differences of spatio-temporal predictions and observed daily mean PM_{10} concentrations using the sum-metric covariance model with the closest (approx. strongest correlated) 50 neighbouring spatio-temporal locations.
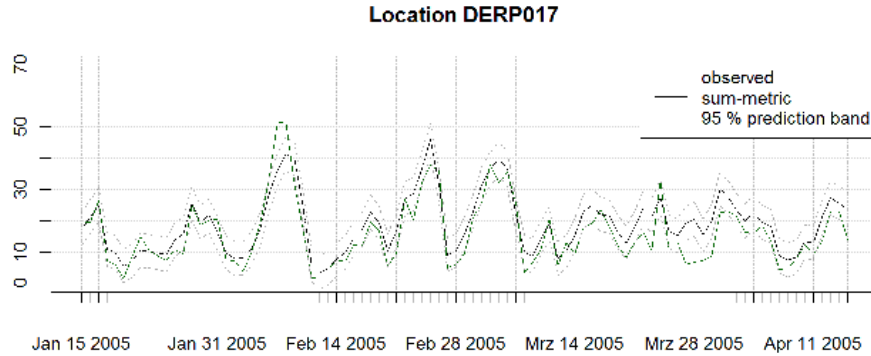

Figure 7: Subset of the time series of observed and predicted PM_{10} at a single station in Lower Saxony along with its 95 % prediction intervals.

stations with an RMSE of 10 and larger correspond to the locations with the largest annual mean concentrations ($> 22 \ \mu g/m^3$).

The added value of spatio-temporal kriging lies in the flexibility of the model. We are now in the position to not only interpolate at unobserved locations in space, but also at unobserved time instances. This makes spatio-temporal kriging a suitable tool to fill gaps in time series not only based on the time series solely, but also including some of its spatial neighbours. A very irregular sampled data set would as well largely benefit from a spatio-temporal approach, as spatially close but unobserved locations in one time slice are not of any help in a purely spatial approach, but the spatio-temporal model would benefit from the observed value nearby at another time instance. In a completely regular data set, the distance to a spatio-temporal neighbour is at least as large as the pure spatial distance and hence the correlation is weaker. Furthermore, being able to capture the covariance structure over space and time might foster a better understanding of the process under study.

While we see spatio-temporal modelling being a powerful tool, the cross-validation results in Table 3 show that spatio-temporal kriging will not solve the problem of all poorly spatially captured phenomena. Further preprocessing steps might be necessary to improve the modelling of this PM_{10} data set such as for instance a temporal AR-model followed by spatio-temporal residual kriging or

Table 3: Leave-one-out cross-validation results. The column wMSE refers to the optimised value from the variogram estimation.

covariance model	wMSE	neigh.	RMSE	MAE	ME	COR
pure Spatial		10	6.15	4.09	-0.01	0.84
separable	[6.82]	10	6.08	4.04	-0.01	0.84
product-sum	[6.91]	10	6.08	4.04	-0.01	0.84
metric	[10.05]	10	6.11	4.07	0.03	0.84
sum-metric	[3.31]	10	6.16	4.08	-0.06	0.84
simple sum-metric	[3.31]	10	6.14	4.08	-0.02	0.84
pure Spatial		50	6.10	4.07	0.00	0.84
separable	[6.82]	50	6.05	4.04	0.01	0.84
product-sum	[6.91]	50	6.05	4.04	0.00	0.84
metric	[10.05]	50	6.07	4.08	0.03	0.84
sum-metric	[3.31]	50	6.14	4.09	-0.01	0.84
simple sum-metric	[3.31]	50	6.14	4.08	-0.02	0.84

using further covariates in a preceding (linear) modelling step. Providing the best possible model of PM_{10} concentrations across Germany was beyond the scope of this paper.

The selection of a spatio-temporal covariance model should not only be made based on the (weighted) mean squared difference between the empirical and model variogram surfaces (presented in Table 2), but also on conceptional choices and visual (Figure 3) judgement of the fits. Even though the function `fit.StVariogram` provides optimisation routines to fit different spatio-temporal variogram models, the numerical routines in the background may struggle to find the optimal parameters. Besides the lower and upper boundaries of the parameter space, the control parameter `parscale` of the `optim` function is a valuable option to improve the convergence of the optimisation. With passing `parscale` as entry of the list `control` a vector of scalars must be passed that controls the different levels of magnitude of the variogram parameters. In most applications, a change of 1 in the sills will have a stronger influence on the variogram surface than a change of 1 in the ranges. The problem becomes more difficult with an increasing number of parameters. In our application, using the simple sum-metric model as starting values for the full sum-metric model improved the convergence speed of the more complex model. In the presented application, the sum-metric models turns out to be the same as the simple sum-metric model. While this might at first sight be due to using the simpler model to generate starting values, different non simplified starting models converged to the same result.

Generally, it is important to keep in mind the strong interaction of the model parameters. It is typically not easy to distinguish how much of the spatio-temporal nugget and sill is attributed to spatial, temporal or joint components. In this paper we considered a joint numerical approach, but step-wise approaches where the components are estimated separately could as well be considered. The interested reader is also referred to Nash [2014]. However, all optimisation approaches follow the premise that the studied process can be approximated with the given model and available data. If this premise fails, no optimal model can be selected.

An extension towards a restricted maximum likelihood method (REML) to fit the spatio-temporal variogram model would be desirable, as it overcomes some of the above mentioned drawbacks of the method of moments based approaches and would additionally provide standard errors for the parameter estimates. A REML approach would allow to take into account that sample variogram values are correlated. However, for large datasets (as in the spatio-temporal case), it is computationally more feasible to use a least squares fitting. To reduce the correlation of the variogram values, some randomisation could be implemented in large data sets, to calculate the sample variogram based on partially overlapping or even disjoint sets of observations.

The selected anisotropy as proxy to the relation of spatial and temporal distance in determining the strongest correlated neighbours might show a distortion for some models when only few neighbours are used towards the true set of the most correlated locations. However, this effect vanishes as soon as the spatio-temporal range of the model is sufficiently represented through the set of nearest neighbours.

As mentioned by Kyriakidis and Journel [1999], an alternative to space-time kriging might be co-kriging. However, this is only feasible if the number of time replicates is (very) small, as the number of cross variograms to be modelled equals the number of *pairs* of time replicates. Also, co-kriging can only interpolate for these time slices, and not inbetween or beyond them. It does however provide prediction error covariances, which can help assessing the significance of estimated *change* parameters [Pebesma et al., 2005, Pebesma and Duin, 2005]. Several of the space-time

variograms presented here may be approximated by sets of direct variograms and cross-variograms.

Fitting variogram models to sample space-time variograms is in our implementation done by `stats::optim`. Our example script uses method `L-BFGS-B` and provides upper and lower parameter boundaries, e.g. to make sure sill parameters do not become negative. There has been a lot of research in optimization since the methods in `optim`, some of which has been reported in the special issue of the Journal of Statistical Software [Varadhan, 2014], and we do see potential to improve the options in this respect.

The approximate selection of the most correlated neighbours solves the lack of a natural notion of a joint distance across space and time. However, other sampling properties might introduce a bias in the prediction. The prediction at an unobserved location with a cluster of observations at one side will be biased towards this cluster and neglect the locations towards the other directions. Similar as the quadrant search in the pure spatial case an octant wise search strategy for the local neighbourhood would solve this limitation. A simpler stepwise approach to define an n-dimensional neighbourhood might already be sufficient in which at first n_s spatial neighbours and then from each spatial neighbour n_t time instances are selected, such that $n_s \cdot n_t \approx n$.

The presented example considers stationary random fields that are isotropic in space. Further extensions towards more sophisticated variogram estimations allowing also for spatial geometric anisotropy are desirable. One could for instance plot variogram maps for spatial separation in North and South direction for each temporal lag. However, the current implementation does not allow to use the anisotropy parameter `anis` of the pure spatial variogram definition. Nevertheless, a preliminary rescaling of coordinates would be a possible workaround. This route has for instance been taken by Gasch et al. [2015] performing 3D+T kriging . The soil profiles in their study show a clear difference in horizontal and vertical variography. To correct for this, the depth dimension of the data has been rescaled to correspond with the dimensions of the horizontal distances before hand. In the subsequent study, these pseudo 3D coordinates have been used to fit the spatio-temporal variograms and perform kriging.

The code in model definitions is meant to be kept both flexible and simple. This is based on i) re-producing the notion of the geostatistical models in the R code and in ii) reusing existing definitions and functions of the pure spatial cases that have been available for many years in **gstat**. The data handling benefits to a large degree from the implementations in the **spacetime** R package.

5   Conclusions

The spatio-temporal extensions to **gstat** allow to model a set of spatio-temporal covariance functions. The implemented functionality eases estimation, visualisation and understanding of spatio-temporal covariance functions. The extension and reuse of already available function structures and nomenclature facilitates an easy translation of spatial workflows to handle spatio-temporal data. The numerical estimation of the variogram parameters might be tricky and needs a large degree of the users attention. It is advised to carefully check the outcome of the `optim` routine after optimisation. Spatio-temporal kriging predictions can be made in a global and a local neighbourhood set-up, while the latter will be the preferred solution for most spatio-temporal data sets and common computer hardware configurations.

Spatio-temporal covariance structures carry valuable information, but a spatio-temporal model is not guaranteed to outperform pure spatial predictions. The benefit in terms of prediction quality of spatio-temporal kriging becomes only apparent if sufficiently strong correlated locations are added with the temporal dimension (i.e. if the model permits strong correlation across time). Nevertheless, the spatio-temporal covariance model might be of interest in itself.

Besides some publications where the authors of this paper were involved in, such as Kilibarda et al. [2014], the software presented here has proven useful in several independent publications, examples of which are [Marek et al., 2015, Biondi, 2013, Hu et al., 2015, Yoon et al., 2014].

6   Acknowledgements

References

Khandoker Shuvo Bakar and Sujit K Sahu. spTimer: Spatio-temporal bayesian modelling using R. *Journal of Statistical Software*, 63(15):1–32, 2015. URL https://dx.doi.org/10.18637/jss.v063.i15.

Alina Beygelzimer, Sham Kakadet, John Langford, Sunil Arya, David Mount, and Shengqiao Li. *FNN: Fast Nearest Neighbor Search Algorithms and Applications*, 2013. URL https://CRAN.R-project.org/package=FNN. R package version 1.1.

Richard A. Bilonick. Monthly hydrogen ion deposition maps for the northeastern U.S. from July 1982 to September 1984. *Atmospheric Environment (1967)*, 22(9):1909 – 1924, 1988. ISSN 0004-6981. URL https://www.sciencedirect.com/science/article/pii/0004698188900807.

F. Biondi. Space-time kriging extension of precipitation variability at 12 km spacing from tree-ring chronologies and its implications for drought analysis. *Hydrology and Earth System Sciences Discussussions*, 10:4301–4335, 2013. URL https://dx.doi.org/10.5194/hessd-10-4301-2013.

Noel Cressie and Christopher K Wikle. *Statistics for spatio-temporal data*. Wiley, 2011.

L. De Cesare, D.E Myers, and D Posa. Estimating and modeling space-time correlation structures. *Statistics & Probability Letters*, 51(1):9–14, 2001. ISSN 0167-7152. URL https://dx.doi.org/10.1016/S0167-7152(00)00131-0.

S. De Iaco, D.E. Myers, and D. Posa. Space-time analysis using a general product-sum model. *Statistics & Probability Letters*, 52(1):21–28, 2001. ISSN 0167-7152. URL https://dx.doi.org/10.1016/S0167-7152(00)00200-5.

Andrew O Finley, Sudipto Banerjee, and Alan E Gelfand. spBayes for large univariate and multivariate point-referenced spatio-temporal data models. *Journal of Statistical Software*, 63(13), 2015. URL https://dx.doi.org/10.18637/jss.v063.i13.

Caley K. Gasch, Tomislav Hengl, Benedikt Gräler, Hanna Meyer, Troy S. Magney, and David J. Brown. Spatio-temporal interpolation of soil water, temperature, and electrical conductivity in 3D + T: The Cook agronomy farm data set. *Spatial Statistics*, 14, Part A:70 – 90, 2015. ISSN 2211-6753. URL https://www.sciencedirect.com/science/article/pii/S2211675315000251.

Yi Hu, Rui Li, Robert Bergquist, Henry Lynn, Fenghua Gao, Qizhi Wang, Shiqing Zhang, Liqian Sun, Zhijie Zhang, and Qingwu Jiang. Spatio-temporal transmission and environmental determinants of schistosomiasis japonica in anhui province, china. *PLoS Neglected Tropical Diseases*, 9(2), 2015. URL https://dx.doi.org/10.1371/journal.pntd.0003470.

Milan Kilibarda, Tomislav Hengl, Gerard B. M. Heuvelink, Benedikt Gräler, Edzer Pebesma, Melita Perčec Tadić, and Branislav Bajat. Spatio-temporal interpolation of daily temperatures for global land areas at 1 km resolution. *Journal of Geophysical Research: Atmospheres*, 119(5):2294–2313, 2014. ISSN 2169-8996. URL https://dx.doi.org/10.1002/2013JD020803.

Phaedon C. Kyriakidis and André G. Journel. Geostatistical space-time models: A review. *Mathematical Geology*, 31(6):651–684, 1999. URL https://dx.doi.org/10.1023/A:1007528426688.

Finn Lindgren and Håvard Rue. Bayesian spatial modelling with R-INLA. *Journal of Statistical Software*, 63(19), 2015. URL https://dx.doi.org/10.18637/jss.v063.i19.

Lukáš Marek, Pavel Tuček, and Vít Pászto. Using geovisual analytics in Google Earth to understand disease distribution: a case study of campylobacteriosis in the Czech Republic (2008–2012). *International Journal of Health Geographics*, 14(7):1–13, 2015. URL https://link.springer.com/article/10.1186/1476-072X-14-7.

John C. Nash. On best practice optimization methods in R. *Journal of Statistical Software*, 60(2):1–14, 2014. URL https://www.jstatsoft.org/v60/i02.

Edzer Pebesma. spacetime: Spatio-temporal data in R. *Journal of Statistical Software*, 51(7):1–30, 2012. URL https://www.jstatsoft.org/v51/i07/.

Edzer J. Pebesma. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30:683–691, 2004. URL https://dx.doi.org/10.1016/j.cageo.2004.03.012.

Edzer J. Pebesma and Richard N.M. Duin. Spatio-temporal mapping of sea floor sediment pollution in the North Sea. In Philip Renard and Roland Froidevaux, editors, *Fifth European Conference on Geostatistics for Environmental Applications, GeoENV2004*, pages 367–378. Springer, 2005. URL https://dx.doi.org/10.1007/3-540-26535-X_31.

Edzer J. Pebesma, Richard N.M. Duin, and Peter A. Burrough. Mapping sea bird densities over the North Sea: Spatially aggregated estimates and temporal changes. *Environmetrics*, 16(6):573–587, 2005. URL https://dx.doi.org/10.1002/env.723.

Martin Schlather, Alexander Malinowski, Peter J. Menck, Marco Oesting, and Kirstin Strokorb. Analysis, simulation and prediction of multivariate random fields with package RandomFields. *Journal of Statistical Software*, 63(8):1–25, 2014. URL https://www.jstatsoft.org/v63/i08.

Fabio Sigrist, Hans R. Künsch, and Werner A. Stahel. spate: An R package for spatio-temporal modeling with a stochastic advection-diffusion process. *Journal of Statistical Software*, 63(14): 1–23, 2015. URL https://www.jstatsoft.org/v63/i14/.

J.J.J.C Snepvangers, G.B.M Heuvelink, and J.A Huisman. Soil water content interpolation using spatio-temporal kriging with external drift. *Geoderma*, 112(3 − 4):253–271, 2003. ISSN 0016-7061. URL https://dx.doi.org/10.1016/S0016-7061(02)00310-5.

Ravi Varadhan. Numerical optimization in R: Beyond optim. *Journal of Statistical Software*, 60 (1):1–3, 9 2014. URL https://dx.doi.org/10.18637/jss.v060.i01.

Seo Youn Yoon, Srinath K. Ravulaparthy, and Konstadinos G. Goulias. Dynamic diurnal social taxonomy of urban environments using data from a geocoded time use activity-travel diary and point-based business establishment inventory. *Transportation Research Part A: Policy and Practice*, 68 (0):3 − 17, 2014. ISSN 0965-8564. URL https://dx.doi.org/10.1016/j.tra.2014.01.004.

Benedikt Gräler
Institute for Geoinformatics, University of Münster
Heisenbergstr. 2, 48149, Münster
Germany
ben.graeler@uni-muenster.de

Edzer Pebesma
Institute for Geoinformatics, University of Münster
Heisenbergstr. 2, 48149, Münster
Germany
edzer.pebesma@uni-muenster.de

Gerard Heuvelink
Department of Environmental Sciences, Wageningen University
PO Box 47, 6700AA, Wageningen
The Netherlands
gerard.heuvelink@wur.nl

7 Appendix

7.1 Derivation of the separable covariance and variogram identities

The separable covariance and variogram identity is readily available through

$$
\begin{aligned}
C_{\text{sep}}(h,u) &= C_{\text{s}}(h)C_{\text{t}}(u) = sill \cdot \bar{c}_s(h)\bar{c}_t(u) \\
\gamma_{\text{sep}}(h,u) &= C_{\text{sep}}(0,0) - C_{\text{sep}}(h,u) \\
&= sill\left(1 - \bar{c}_s(h) \cdot \bar{c}_t(u)\right) \\
&= sill\left(1 - \left(1 - \bar{\gamma}_s(h)\right)\left(1 - \bar{\gamma}_t(u)\right)\right) \\
&= sill\left(1 - \left(1 - \bar{\gamma}_s(h) - \bar{\gamma}_t(u) + \bar{\gamma}_s(h)\bar{\gamma}_t(u)\right)\right) \\
&= sill\left(\bar{\gamma}_s(h) + \bar{\gamma}_t(u) - \bar{\gamma}_s(h)\bar{\gamma}_t(u)\right)
\end{aligned}
$$

where $\bar{c}$ and $\bar{\gamma}$ are normalised correlation and correlogram functions respectively.

7.2 Derivation of the product-sum covariance and variogram identities

The product-sum covariance and variogram identity is readily available through:

$$
\begin{aligned}
C_{\text{ps}}(h,u) &= k \cdot C_{\text{s}}(h)C_{\text{t}}(u) + C_{\text{s}}(h) + C_{\text{t}}(u) \\
\gamma_{\text{ps}}(h,u) &= C_{\text{ps}}(0,0) - C_{\text{ps}}(h,u) \\
&= k \cdot C_{\text{s}}(0)C_{\text{t}}(0) + C_{\text{s}}(0) + C_{\text{t}}(0) \\
&\quad - \left(k \cdot C_{\text{s}}(h)C_{\text{t}}(u) + C_{\text{s}}(h) + C_{\text{t}}(u)\right) \\
&= k \cdot \text{sill}_{\text{s}} \cdot \text{sill}_{\text{t}} + \text{sill}_{\text{s}} + \text{sill}_{\text{t}} \\
&\quad - k \cdot \left[\left(\text{sill}_{\text{s}} - \gamma_{\text{s}}(h)\right)\left(\text{sill}_{\text{t}} - \gamma_{\text{t}}(u)\right)\right] - \left(\text{sill}_{\text{s}} - \gamma_{\text{s}}(h)\right) - \left(\text{sill}_{\text{t}} - \gamma_{\text{t}}(u)\right) \\
&= k \cdot \text{sill}_{\text{s}} \cdot \text{sill}_{\text{t}} + \text{sill}_{\text{s}} + \text{sill}_{\text{t}} \\
&\quad - k \cdot \left[\text{sill}_{\text{s}} \cdot \text{sill}_{\text{t}} - \text{sill}_{\text{s}} \cdot \gamma_{\text{t}}(u) - \text{sill}_{\text{t}} \cdot \gamma_{\text{s}}(h) + \gamma_{\text{s}}(h)\gamma_{\text{t}}(u)\right] \\
&\quad - \text{sill}_{\text{s}} + \gamma_{\text{s}}(h) - \text{sill}_{\text{t}} + \gamma_{\text{t}}(u) \\
&= k \cdot \text{sill}_{\text{t}}\gamma_{\text{s}}(h) + k \cdot \text{sill}_{\text{s}}\gamma_{\text{t}}(u) - k\gamma_{\text{s}}(h)\gamma_{\text{t}}(u) + \gamma_{\text{s}}(h) + \gamma_{\text{t}}(u) \\
&= (k \cdot \text{sill}_{\text{t}} + 1)\gamma_{\text{s}}(h) + (k \cdot \text{sill}_{\text{s}} + 1)\gamma_{\text{t}}(u) - k\gamma_{\text{s}}(h)\gamma_{\text{t}}(u)
\end{aligned}
$$