



Proposal/Contract no.: 033811
Project start: September 1, 2006
Project end: August 31, 2009



INTAMAP

Interoperability and Automated Mapping

SIXTH FRAMEWORK PROGRAMME

PRIORITY IST-2005-2.5.12

ICT for Environmental Risk Management

Deliverable 3.8

Integrated R codes for copula-based geostatistics

Title of Deliverable	Integrated R codes for copula-based geostatistics
Deliverable reference number	INTAMAP D3.8
Related WP and Tasks	WP3, Task 3.7
Type of Document	Public
Authors	KLU
Date	28 July 2009
Version	1.0

Project coordinator

Prof. Dr. Edzer J. Pebesma

Utrecht University, The Netherlands

E-mail: e.pebesma@geo.uu.nl

<http://www.intamap.org/>

Revision History

Version	Date	Changes	Authors
1.0	28-07-2009	First draft	Hannes Kazianka

Related task

Task 3.7 Development of computer code (R) and testing

This task is crucial for the operational deployment of the developed methodology for outlying, non-Gaussian and extreme-valued distributions in an automated mapping system. Work on this task will begin relatively early in the project, in month 6, and will be completed until the end of the second year of the project (month 24), in order to allow sufficient time for developing code for monitoring network optimization methods based on the interpolation methods developed in tasks 3.1 - 3.6. The numerical accuracy, performance and stability of the methods and algorithms will be tested with both real and synthetic data to evaluate their potentials and limitations.

Active partners: KLU

Legal Notices

The information in this document is subject to change without notice. The Members of the INTAMAP Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the INTAMAP Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Contents

1	Introduction	2
2	Copula-based geostatistics	2
2.1	Copula-based spatial modeling	2
2.2	The Gaussian spatial copula model	3
2.3	Copula-based spatial interpolation	3
3	Copula-based spatial interpolation algorithms integrated in the <code>intamap</code> R-library	4
3.1	Functionality	4
3.2	The <code>interpolate</code> function	6
3.3	Details on the implementation	7
4	Applications	9
4.1	Helicopter data	9
4.2	Scenario 6 (“worst case”)	11
5	Conclusions	13
A	Help files of the <code>spatialCopula</code> R-library	15
	<code>spatialCopula-package</code>	15
	<code>estimateParameters.copula</code>	16
	<code>copulaEstimation</code>	17
	<code>spatialPredict.copula</code>	20
	<code>bayesCopula</code>	21
	<code>intamapExampleObject</code>	23

Executive Summary

The aim of this deliverable is to describe the R-code for the copula-based spatial methodology that has been developed within the INTAMAP project. Moreover, it presents results for both a real and a simulated data set that are obtained using the implemented algorithms.

Copulas provide a way to separately specify the dependence structure and the univariate marginal distributions of a multivariate distribution. This makes it possible to define flexible classes of multivariate distributions that are non-Gaussian. In Task 3.3 and 3.4 of the INTAMAP project we have developed a framework to employ copula functions in geostatistics. To work with the proposed methodology on data sets we have developed R-code that is fully integrated in the `intamap` R-library. Towards the end of the INTAMAP project this library will be uploaded to the Comprehensive R Archive Network (CRAN) (<http://cran.r-project.org>), from where users can freely download the code. The whole `intamap` R-library, including the part that deals with the copula-based analysis, will be maintained and possibly extended by its developers after INTAMAP finishes.

The strong side of geostatistical methods that make use of copulas is that they are able to deal with data that arise from hazards and emergencies. Choosing the univariate marginals appropriately makes it possible to handle extreme data better than classical models, such as kriging, do. This fact is demonstrated by a comparative study on the helicopter data set and the Scenario 6 data set (combined European "worst case"-scenario) which are both provided by the project partner Bundesamt für Strahlenschutz (BfS).

1 Introduction

Within the INTAMAP project novel statistical tools for spatial modeling and interpolation have been developed. Among them is the copula-based methodology which is designed for dealing with non-Gaussian and extreme value data. This report is focused on the implementation of this method in the R statistical software. R is a free and open source programming and statistical computing language that can be downloaded from <http://www.r-project.org>. The developed code is integrated in the `intamap` R-library that serves both as a back-end of the INTAMAP web-based interpolation service and for interactive use within the R environment.

The report is organized as follows. In Section 2 we discuss copula-based geostatistical techniques. Section 3 describes how this methodology is implemented in R and how it can be applied within the R user interface. Section 4 presents results for two data sets that are obtained using the proposed algorithms while Section 5 is devoted to conclusions.

2 Copula-based geostatistics

This section briefly describes the theory behind spatial modeling and interpolation with the help of copula functions. More information can be found in the INTAMAP -Deliverables D3.2 and D3.4.

2.1 Copula-based spatial modeling

Assume that $\{Z(\mathbf{x}) \mid \mathbf{x} \in \mathcal{S}\}$ is a second-order stationary random field where $\mathcal{S} \subseteq \mathbb{R}^2$ is the area of interest, and suppose we have a single realization $\mathbf{Z} = (Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n))^T$ of this field where $\mathbf{x}_1, \dots, \mathbf{x}_n$ are distinct observation locations. Bardossy (2006) in his pioneering work presented a method for spatial modeling using copulas that aims to describe all multivariate distributions of the random field with the help of copulas. However, since his model only parameterizes the dependence structure of the copula (the parameters are later called $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$), we use the formulation developed in Kazianka and Pilz (2009a). Let $F_{\boldsymbol{\eta}}$ denote the univariate distribution of the random process, where $\boldsymbol{\eta}$ are the corresponding parameters. With the help of Sklar's Theorem (Nelsen, 2006) we are able to model the relation between $Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n)$ by

$$P(Z(\mathbf{x}_1) \leq z_1, \dots, Z(\mathbf{x}_n) \leq z_n) = C_{\boldsymbol{\lambda}, \boldsymbol{\theta}}(F_{\boldsymbol{\eta}}(z_1), \dots, F_{\boldsymbol{\eta}}(z_n)),$$

where $C_{\boldsymbol{\lambda}, \boldsymbol{\theta}}$ denotes a multivariate, continuous copula as a function of the correlation parameters $\boldsymbol{\theta}$ and copula specific parameters $\boldsymbol{\lambda}$. In this model $C_{\boldsymbol{\lambda}, \boldsymbol{\theta}}$ describes the spatial dependence and is therefore called a spatial copula. Note that spatial copulas describe spatial dependence over the whole range of quantiles and not only the mean squared dependence as the variogram does. For details on parameter estimation and on incorporation of covariates, e.g. a spatial trend, consult Kazianka and Pilz (2009b).

Not all continuous copulas are suitable for geostatistical modeling. A natural assumption for a spatial copula is symmetry, implying that, for example, the dependence between locations \mathbf{x}_1 and \mathbf{x}_2 is the same as the dependence between \mathbf{x}_2 and \mathbf{x}_1 . In general it means

that $C_{\lambda, \theta}(u_1, \dots, u_n) = C_{\lambda, \theta}(u_{\pi(1)}, \dots, u_{\pi(n)})$ for an arbitrary permutation π and $n \geq 2$. Moreover, if h denotes the distance between two locations, we want to add the following two restrictions: as $h \rightarrow \infty$ we require $C_{\lambda, \theta}(u_1, u_2) \rightarrow u_1 u_2$ which implies that far distant observations are nearly independent, and as $h \rightarrow 0$ we require in the absence of measurement errors that $C_{\lambda, \theta}(u_1, u_2) \rightarrow \min(u_1, u_2)$ ensuring that observations that are very close to each other have a strong dependence.

Frequently used spatial copulas are the non-central χ^2 -copula introduced by Bardossy (2006) and the Gaussian copula presented in the next section. The non-central χ^2 -copula is a flexible, radially asymmetric copula where λ is equal to the squared non-centrality parameter of a non-central χ^2 -distribution. Kazianka and Pilz (2009a) showed that if the Gaussian copula is used, the model is equivalent to the well-known trans-Gaussian kriging model (De Oliveira et al., 1997). For any non-Gaussian copula, e.g. for the non-central χ^2 -copula, the spatial copula model generalizes trans-Gaussian kriging.

2.2 The Gaussian spatial copula model

The most important class of random fields are the Gaussian random fields where all multivariate distributions follow a Gaussian distribution. Let the margins be univariate Gaussian with mean μ and variance σ^2 , hence, $F_{\eta} = \Phi_{\mu, \sigma^2}$. The copula-based spatial model includes the Gaussian random field as a special case which occurs when the spatial copula is equal to the so-called Gaussian copula

$$C_{\Sigma}^G(u_1, \dots, u_n) = \Phi_{\mathbf{0}, \Sigma_{\theta}}(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_n)),$$

where $\Phi_{\mathbf{0}, \Sigma_{\theta}}$ is the distribution function of the multivariate Gaussian distribution with mean vector $\mathbf{0}$ and correlation matrix Σ_{θ} . The restrictions for spatial copulas stated in Section 2.1 are all satisfied and it is even possible to describe negative spatial dependence. The dependence structure of the Gaussian copula is parameterized by assuming that its correlation function follows one of the classical geostatistical models with parameters θ e.g. the Matern model. A copula specific parameter λ does not occur. A restriction of the Gaussian copula is that it models not only a symmetric but even a radially symmetric dependence, where high and low quantiles have equal dependence properties.

If the univariate margins are not Gaussian but stem from a family of continuous distributions F_{η} , the Gaussian spatial copula model is equivalent to trans-Gaussian kriging with transformation function $g_{\eta}(x) = \Phi^{-1}(F_{\eta}(Z(\mathbf{x})))$. In trans-Gaussian kriging, however, most often only one-parameter transformations, e.g. the Box-Cox transformation, are applied which makes it impossible to deal with extreme value or multi-modal data. Using the copula-based approach we simply choose the marginal distribution F_{η} as a generalized extreme value distribution or a mixture of Gaussians to account for these special types of data (Kazianka and Pilz, 2009b; Stöhlker et al., 2009).

2.3 Copula-based spatial interpolation

Let $\mathbf{x}_0 \in \mathcal{S}$ be an unobserved location where prediction should take place. The plug-in predictive density of $Z(\mathbf{x}_0)$ given \mathbf{Z} and parameters $\hat{\Theta} = (\hat{\lambda}, \hat{\theta}, \hat{\eta})$ can be calculated by

using the conditional copula (Kazianka and Pilz, 2009a; Bardossy and Li, 2008)):

$$p(z(\mathbf{x}_0) | \hat{\Theta}, \mathbf{Z}) = c_{\hat{\lambda}, \hat{\theta}}(F_{\hat{\eta}}(z(\mathbf{x}_0)) | \mathbf{Z}) f_{\hat{\eta}}(z(\mathbf{x}_0)).$$

If the Gaussian spatial copula model is used the latter equation simplifies to

$$p^G(z(\mathbf{x}_0) | \hat{\Theta}, \mathbf{Z}) = \frac{\phi_{\mu, \sigma^2}(\Phi^{-1}(F_{\hat{\eta}}(z(\mathbf{x}_0)))) f_{\hat{\eta}}(z(\mathbf{x}_0))}{\phi(\Phi^{-1}(F_{\hat{\eta}}(z(\mathbf{x}_0))))},$$

where ϕ_{μ, σ^2} is a univariate Gaussian density with mean $\mu = \Sigma_{\hat{\theta}}^{12} \Sigma_{\hat{\theta}}^{22^{-1}} \mathbf{a}$ and variance $\sigma^2 = 1 - \Sigma_{\hat{\theta}}^{12} \Sigma_{\hat{\theta}}^{22^{-1}} \Sigma_{\hat{\theta}}^{21}$. Furthermore, $\Sigma_{\hat{\theta}}^{22}$ is the correlation matrix of the values at the known locations, $\Sigma_{\hat{\theta}}^{12} = \Sigma_{\hat{\theta}}^{21^T}$ is the vector of correlations between the values at the known locations and \mathbf{x}_0 and $\mathbf{a} = (\Phi^{-1}(F_{\hat{\eta}}(Z(\mathbf{x}_1))), \dots, \Phi^{-1}(F_{\hat{\eta}}(Z(\mathbf{x}_n))))^T$.

The Bayes estimator for $Z(\mathbf{x}_0)$ under the quadratic loss is the mean of the predictive distribution, $E(Z(\mathbf{x}_0) | \hat{\Theta}, \mathbf{Z})$. It can be calculated by integrating over the unit interval:

$$\begin{aligned} E(Z(\mathbf{x}_0) | \hat{\Theta}, \mathbf{Z}) &= \int_{-\infty}^{\infty} z(\mathbf{x}_0) c_{\hat{\lambda}, \hat{\theta}}(F_{\hat{\eta}}(z(\mathbf{x}_0)) | \mathbf{Z}) f_{\hat{\eta}}(z(\mathbf{x}_0)) dz(\mathbf{x}_0) \\ &= \int_0^1 F_{\hat{\eta}}^{-1}(u(\mathbf{x}_0)) c_{\hat{\lambda}, \hat{\theta}}(u(\mathbf{x}_0) | \mathbf{Z}) du(\mathbf{x}_0). \end{aligned} \quad (1)$$

If the predictive mean does not exist, we suggest to use the median of the predictive distribution as the predictor.

3 Copula-based spatial interpolation algorithms integrated in the intamap R-library

R is a very popular open source statistical software. Applied researchers from different statistical disciplines contribute to the R-project and develop add-on libraries. At the very start of the INTAMAP project it was decided that the implementation of the Web Processing Service (WPS) should be done in R. INTAMAP Deliverable D4.5 describes the developed `intamap` R-library of which the copula-based interpolation method is an integral part. In the following we are going to describe its implementation in more detail.

3.1 Functionality

Previously, the code for the copula-based interpolation was in a separate non-public package called `spatialCopula` due to copyright issues (help files for this package can be found at the end of this report). Since these issues are resolved, all code is now integrated in the `intamap` R-library. Table 1 lists the implemented functions and their objectives. There are two main functions, `estimateParameters.copula` and `spatialPredict.copula`, that perform parameter estimation and spatial prediction for an `intamapObject` of the class `copula`.

```
intamapObject<-createIntamapObject(observations=observations,
predictionLocations=predictionLocations,class="copula")
```

The more experienced user may also find it helpful to work with some of the remaining

<code>estimateParameters.copula</code>	Gets an <code>intamapObject</code> as input and returns an updated object with the parameter estimates. Choice of univariate marginal distribution and starting parameters for the optimization routine is done automatically if needed.
<code>copulaEstimation</code>	Performs maximum likelihood estimation for a given copula-based spatial model and predefined starting parameters.
<code>findmargins</code>	Chooses automatically the “best” fitting family of univariate marginal distributions.
<code>findmarginbounds</code>	Stores the optimization bounds for the marginal distribution families.
<code>normtesting, lnormtesting, gevtesting, ttesting, logistesting</code>	Performs <code>ks.test</code> for a specific family of marginal distributions. Estimates parameters based on maximum likelihood and the independence hypothesis.
<code>gevfit</code>	Advanced maximum likelihood estimation of parameters belonging to the GEV-distribution.
<code>covar, maternmodel</code>	Calculates correlation matrix.
<code>profilelikelihood, profilelikelihood2</code>	Administrates the maximum likelihood estimation. Profiling is used because the number of parameters is typically high.
<code>optimfun1, ..., optimfun7</code>	Only needed because the objective function in the optimization must have the parameters which are optimized as the first argument. Calls <code>likelianiso</code> or <code>likelianiso2</code> .
<code>likelianiso, likelianiso2</code>	Calculates the likelihood for the Gaussian (<code>likelianiso2</code>) or for the χ^2 -copula model (<code>likelianiso</code>).
<code>logdmvnormnoconst</code>	Efficient evaluation of the multivariate Gaussian density.
<code>spatialPredict.copula</code>	Gets an <code>intamapObject</code> with estimated parameters as input and returns an updated object with predictions.
<code>bayesCopula, predictioncopula</code>	Performs spatial prediction at given locations.
<code>condcopuladensgauss, condcopuladenschi2</code>	Calculates conditional copula for the Gaussian and the χ^2 -copula model.
<code>meanintgauss, meanintchi2</code>	Calculates predictive mean for the Gaussian and the χ^2 -copula model
<code>varintgauss, varintchi2</code>	Calculates predictive variance for the Gaussian copula model.
<code>exceedance</code>	Calculates exceedance probabilities for the Gaussian copula model.
<code>findquantiles</code>	Calculates predictive quantiles for the Gaussian copula model.

Table 1: Implemented functions and their objectives.

functions such as `copulaEstimation` or `bayesCopula` because fine tuning of the algorithms is easier to do there.

The program is able to deal with both the Gaussian spatial copula model and the χ^2 -copula model and performs maximum likelihood estimation of all model parameters: $\boldsymbol{\eta}$, $\boldsymbol{\theta}$, $\boldsymbol{\lambda}$. It is possible to work with trend surface models by setting the argument `formulaString` of the `intamapObject` accordingly. Moreover, geometric anisotropy can be considered by introducing two parameters a and φ , the anisotropy ratio and the anisotropy angle. All these parameters are optimized simultaneously. The function `bayesCopula` is used for plug-in

prediction at ungauged spatial locations. The name of the function is somewhat misleading. No Bayesian approach is implemented so far, however, it is planned to integrate Bayesian analysis in future versions of the code. It is possible to calculate numerically the predictive mean and variance for the Gaussian and the χ^2 -copula model. Calculation of exceedance probabilities and predictive quantiles is currently supported only for the Gaussian spatial copula model. The requested prediction types are defined by the argument `outputWhat` of the `intamapObject`:

```
outputWhat = list(mean = TRUE, variance = FALSE, excProb = 10, excprob = 20,
                  quantile = 0.025, quantile = 0.975)
```

3.2 The interpolate function

The `interpolate` function of the `intamap` R-package is designed for automatic spatial modeling and interpolation. Its input are the observations and prediction locations as `SpatialPointsDataFrames` and its output is an `intamapObject` with predictions (at a regular grid if no prediction locations are provided). The function then decides which of the following three interpolation methods to apply: `automap`, `psgp` and `copula`. Since the spatial copula approach is especially useful for non-Gaussian data, this method is only selected if any of the four tests below is `TRUE`:

```
test[1] = length(boxplot.stats(dataObs)$out)/length(dataObs) > 0.1
test[2] = fivenum(dataObs)[3] - fivenum(dataObs)[2] < IQR(dataObs)/3
test[3] = fivenum(dataObs)[4] - fivenum(dataObs)[3] < IQR(dataObs)/3
g = boxcox(dataObs ~ 1, lambda = seq(-2.5, 2.5, len = 101), plotit = FALSE)$y
test[4] = g[71] < sort(g)[91]
```

Test 1 is `TRUE` when more than 10% of the data lie beyond the extremes of the whiskers of the boxplot, i.e. more than 10% of the data are marked as outliers. The second or the third test is `TRUE` when the distance from the 25%- or the 75%-quantile to the median is smaller than a third of the interquartile range. Test 4 is `TRUE` when the likelihood under the Gaussian assumption is smaller than the top ten likelihood values obtained using the transformed Gaussian model, where 101 values of `lambda` in the range between -2.5 and 2.5 are tried. If the tests indicate that the copula-based model should be applied and possible computation time constraints are fulfilled, `interpolate` calls `estimateParameters.copula` and `spatialPredict.copula` where further processing steps take place. When automatic interpolation is desired, typically no starting values for the optimization routine are specified and no family of marginal distributions is preselected. Therefore, we propose the following strategy that is implemented in `estimateParameters.copula`:

1. Let the function `estimateAnisotropy` decide whether to include geometric anisotropy into the analysis. If `doRotation==TRUE` get initial values for the anisotropy parameters φ and a : `angle` and `direction`.
2. Use the function `autofitVariogram` of the `automap` R-library to select a correlation function model and starting parameters for the corresponding parameters. Available models are the Gaussian, the exponential, the spherical and the Matern model. If geometric anisotropy is considered, calculation is based on the rotated locations.

3. To find an appropriate univariate marginal distribution function family test the Gaussian, the log-Gaussian, the Student-t, the generalized extreme value (GEV) and the logistic distribution. For every family under consideration calculate the maximum likelihood estimates under the independence hypothesis for the observations. Compute the Kolmogorov-Smirnov statistics for all distribution families. Take the family which has the smallest test statistic and use the maximum likelihood estimates as the starting values for the parameters in the optimization process.
4. If a spatial trend or covariates are considered, all starting values for the regression parameters (except for the intercept term for which the starting value is already defined) are chosen to be 0.
5. Work with the Gaussian spatial copula model.

Steps 1 and 2 are obvious because both functions `estimateAnisotropy` and `autofitVariogram` are already employed in the `automap` interpolation method to provide estimates for the corresponding parameters. The reason to take step 4 is that all initial parameter values are based on the assumption of a constant trend. In step 5 the Gaussian spatial copula model is chosen because of the savings in computation time compared to the χ^2 -copula model. The decision in step 3 to calculate the Kolmogorov-Smirnov statistic without accounting for the spatial dependence of the data is a bit more crucial and also mainly made because of computational speed. A theoretically sound alternative is to perform a likelihood-ratio test: The likelihood function of the copula-based model is maximized for all different choices of the family of univariate marginal distributions. The family which corresponds to the maximum of all the maximized likelihood values is then selected as the one that fits best. However, this would raise the computing time by a factor of five. A possibility could be not to perform the maximization of the likelihood till the very end but to stop the optimization algorithm when it is unable to reduce the likelihood by a certain amount, say 1. The opportunity to do a likelihood ratio test will be implemented in future versions of the code. Spatial prediction is performed with the nearest 50 observations. The local prediction also saves computation time because it avoids inverting large covariance matrices when calculating the conditional copula.

3.3 Details on the implementation

Currently the code depends on the following R-libraries:

- `mvtnorm`: Provides the multivariate Gaussian and Student-t distribution. It is only used for calculating the density and the conditional density of the χ^2 -copula. Its implementation is very slow, therefore, it is likely to be replaced in future versions of the code.
- `MASS`: Provides many statistical tools. The function `fitdistr` is used to calculate starting parameters for the maximum-likelihood estimation.
- `evd`: Provides the generalized extreme value distribution (GEV).

The `evd` R-library provides a function `fgev` that implements maximum likelihood estimation of the parameters of the GEV distribution under the independence hypothesis for the

observations. However, this function seems to be very unstable. Starting values for the maximum likelihood estimation of the copula model should certainly be best possible. If a Kolmogorov-Smirnov test is performed to compare different marginal distribution function families, bad parameter estimates distort the test statistics and may lead on to picking the wrong family. Therefore, we implemented our own function, `gevfit`, for doing maximum likelihood estimation which produces more accurate results.

Because of the large number of variables that need to be optimized when performing maximum likelihood estimation for the spatial copula model, a profile-likelihood approach is used. The parameters of the univariate marginal distribution, η , the correlation function parameters, θ , the copula-specific parameters, λ , and the parameters for geometric anisotropy, φ and a , are optimized in turn. The optimization function in R is `optim` which performs a box-constraint BFGS algorithm. The input variable `tol` of the `copulaEstimation` function serves as a termination condition for this optimization. If the profile-likelihood cannot be reduced by `tol*(abs(actualLikelihoodValue)+tol)` after all variables are updated, the optimization is stopped. The default value for `tol` is 0.001. Although convergence to a global optimum is not assured the profile-likelihood method makes it less likely that the optimization gets stuck in a local optimum. If there is enough time, it is advisable to check the output of `estimateParameters.copula` by trying different starting values for the optimization. These can be set by the argument `copulaParams` in an `intamapObject` of the class `copula`. `copulaParams` is a list that contains the elements `margin`, `correlation`, `trend` and `anisotropy` which are described in the help pages of the function `copulaEstimation` at the end of this report. If the `params$debug.level` argument of the `intamapObject` is set to a value higher than 0, tracing information on the progress of the optimization is produced which can help in troubleshooting.

As discussed in Section 2.3 it is not guaranteed that the mean of the predictive distribution always exists. The only indicator for a non-finite first moment is that the numerical integration in Eq. (1) fails to produce sensible values. If the input variable `testMean` of the `bayesCopula` function is TRUE, we check the predicted values in the following way:

```

index<-!is.na(IntamapObject$predictionLocations@coords[,1]) &
      !is.na(IntamapObject$predictionLocations@coords[,2])
med=median(intamapObject$observations@data$value)
ma=max(intamapObject$observations@data$value)
mi=min(intamapObject$observations@data$value)
if((sum(is.na(prediction$mean[index]))>0 ||
     max(prediction$mean[index])>ma+2*(ma-med) ||
     min(prediction$mean[index])<mi-2*(med-mi))){
warning("Problem in bayesCopula. Estimated mean values are nonsensical.
        Calculating median instead.", call. = FALSE, immediate. = TRUE)
prediction$mean<-predictioncopula(IntamapObject$predictionLocations,
                                IntamapObject$observations,estimates,
                                search,list(mean=FALSE,variance=FALSE,quantiles=0.5),
                                IntamapObject$params$debug.level)$quantiles
prediction$quantile0.5<-prediction$mean
}

```

If any of the predictions is NA or if predicted values are too large or too low, the median of the predictive distribution is calculated and replaces the mean estimate. If the median has already been obtained, then, of course, no re-calculation is done.

Computation time is a major issue for the spatial copula algorithms. The following will influence the computational load:

- The number of observations, n , is the key quantity as far as computation time is concerned. At every optimization step we need to calculate the actual correlation matrix which is of dimension $n \times n$. Furthermore, to evaluate the likelihood there is the need to invert the correlation matrix.
- If the χ^2 -copula model is used instead of the Gaussian spatial copula model, computation will take much longer since a composite-likelihood approach is used (Kazianka and Pilz, 2009b).
- If the Matern model is chosen as the correlation model, the algorithm will be slower because the `besselk` function needs to be repeatedly evaluated and there is one additional parameter.
- If the GEV distribution is chosen as the univariate marginal distribution, the algorithm will be slower because there are three parameters to optimize instead of only two.
- Additionally accounting for covariates slows down the estimation process, since regression parameters are introduced and need to be optimized.
- Prediction is slower when quantiles of the predictive distribution are requested. This is because we need to solve an integral equation numerically.
- Selecting “good” initial values for the optimization reduces the computational load since fewer iterations are needed to reach convergence.
- If the value for `tol` is small, the optimization will be slower but the estimation will be more accurate. If `tol` is large, the opposite is true.

4 Applications

We have discussed in Section 2 that copula-based geostatistical models offer a way to overcome the Gaussian assumption and to work with extreme value data that often occur in environmental applications. In this section we present the results obtained when this methodology is applied to two data sets from the Bundesamt für Strahlenschutz (BfS). Calculation is done in R.

4.1 Helicopter data

The Helicopter data set contains 902 real measurements of gamma dose rates at the locations displayed as blue dots in Figure 1(a). As can be seen from the summary statistics in Table 2 the observed values are very right-skewed and have a high variance. A 3D representation of the observations is given in 1(b) and it shows that there is one hot spot with values about 10 times higher than the background radiation. These facts motivate us to use a

distribution with heavy tails to model the data. We choose the generalized extreme value (GEV) distribution as the univariate marginal distribution. This distribution family is well-known in extreme value theory and generalizes the Gumbel, the Fréchet and the Weibull families. The density function of the GEV distribution is given by

$$f_{GEV}(x) = \begin{cases} \frac{1}{\sigma} \left(1 + K \frac{x-\mu}{\sigma}\right)^{-\frac{1}{K}-1} \exp\left[-\left(1 + K \frac{x-\mu}{\sigma}\right)^{-\frac{1}{K}}\right], & 1 + K \frac{x-\mu}{\sigma} > 0, \\ 0, & \text{otherwise,} \end{cases}$$

where $\mu \in \mathbb{R}$ is the location parameter, $\sigma > 0$ is the scale parameter and $K \in \mathbb{R}$ is the shape parameter. As the spatial copula we choose the Gaussian copula because for 902 observations even the composite likelihood approach to parameter estimation in case of the χ^2 -copula would take too much time. As can be deduced from Figure 1(a) observations with high values (red circles) show geometric anisotropy. Therefore, we include the two additional anisotropy parameters, φ and a , in our model. When we plot the measurements against the coordinates (not shown here), we find out that there is no spatial trend that we need to account for. To be as flexible as possible we use the Matern correlation model with nugget parameter ϑ_1 , range parameter ϑ_2 and smoothness parameter κ to parameterize the dependence structure of the Gaussian copula. In total we have 8 parameters in our model.

	n	Min	Mean	Median	Max	Stand. dev.	Skewness
Helicopter data	902	313	863.9169	780.5	5420	482.0617	4.7398

Table 2: Helicopter data: Statistics of the observed values.

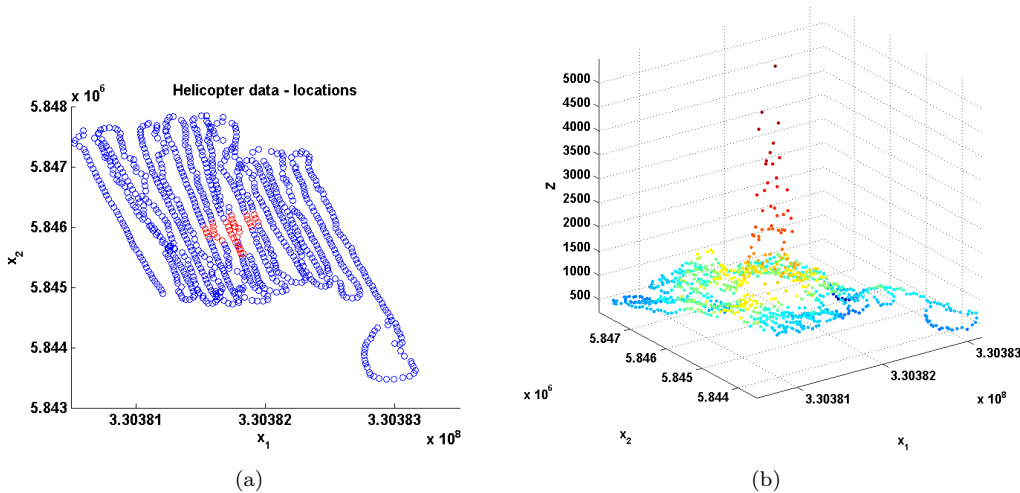


Figure 1: Helicopter data: Observation locations are shown in (a). Locations with measured values larger than 2000 are highlighted. A 3D representation of the data set is given in (b).

Maximum likelihood estimates are given in Table 3. The estimate for the smoothness parameter κ indicates that the underlying random field is not too rough but also not overly smooth.

Parameters	μ	σ	K	ϑ_1	ϑ_2	κ	φ	a
Estimates	776.663	247.329	0.185	0.008	515.960	1.611	1.6767	1.333

Table 3: Helicopter data: Maximum likelihood estimates.

As there are no prediction locations with known measurement values available to judge the predictive performance of our model, we do leave-one-out cross-validation (`nmax=50`). To quantitatively compare our results with those for `idw`, `automap`, `psgp` and `transGaussian`, we calculate the root mean squared error (RMSE) and the mean absolute error (MAE). Undoubtedly, the copula-based model performs best. Because of the use of the GEV distribution as the univariate marginal it is able to model both the background radiation and the extreme observations adequately. Not surprisingly, the second best method in this study is `transGaussian`, since it also performs a transformation of the data. `psgp` leads to smaller cross-validation RMSE and MAE than `automap` or `idw`.

CV-criterion	<code>copula</code>	<code>idw</code>	<code>automap</code>	<code>psgp</code>	<code>transGaussian</code>
RMSE	72.38	167.15	149.57	104.07	85.90
MAE	34.74	63.19	55.95	41.73	37.69

Table 4: Helicopter data: Cross-validation results for different spatial interpolation techniques.

4.2 Scenario 6 (“worst case”)

The Scenario 6 data set contains 3568 simulated measurements of gamma dose rates. The data are right-skewed and have a very high variance. Therefore, we again use the GEV distribution as the univariate marginal distribution for the Gaussian spatial copula model. Since the data set contains extreme hot spots, we expect the underlying random field to be rough. Therefore, we again use the Matern correlation function to parameterize the dependence structure of the Gaussian copula. Geometric anisotropy is also considered.

	n	Min	Mean	Median	Max	Stand. dev.	Skewness
Scenario 6 data	3568	30	3145.19	133	34593.2	5615.95	2.26

Table 5: Scenario 6 data: Statistics of the observed values.

The maximum likelihood estimates for the 8 variables are given in Table 6. As expected, the estimate for the smoothness parameter κ is lower than 0.5, indicating a very rough underlying random field. Note that the marginals of the random field have no finite variance.

Parameters	μ	σ	K	ϑ_1	ϑ_2	κ	φ	a
Estimates	39.105	9.788	0.611	0.0523	24.9874	0.418	0	3.660

Table 6: Scenario 6 data: Maximum likelihood estimates.

Cross-validation results can be obtained from Table 7. There are numerical problems when evaluating the predictive mean (Eq. (1)), indicating that the first moment of the predictive distribution does not exist. Therefore, we took the predictive median as the point predictor for the copula-based model. `automap` gives best results for RMSE, while `copula` outperforms all the other methods in terms of MAE. Generally, in case of extreme value data the MAE is a better quantitative measure for assessing the goodness-of-fit than the RMSE which is supported by the theory of robust statistics (Huber and Ronchetti, 2009).

The difference in predictive performance between `copula` and `automap` is difficult to explain. Figure 2 plots the observed against the predicted values for both methods. The anisotropy

CV-criterion	idw	automap	copula
RMSE	1847.729	1667.711	1902.200
MAE	859.731	719.892	639.5676

Table 7: Scenario 6 data: Cross-validation results for different spatial interpolation techniques.

parameters and correlation function parameters are chosen to be $\vartheta_1 = 0.08$, $\vartheta_2 = 1.99$, $\kappa = 5$, $\varphi = -0.1738$ and $a = 1.64$ in the **automap** method. Although 5 seems to be a strange value for the smoothness parameter in an extreme scenario and the range is unexpectedly short, some of the very high observed values are predicted more accurately. The wrong prediction of observation number 2784 alone adds ≈ 106 to the RMSE for **copula**. The reason for the bad prediction is that after the geometric anisotropy rotation observation number 2928 with a measured value of 100 is the second nearest neighbor when **copula** is used. For **automap** observation number 2928 is only the ninth nearest neighbor and all the other neighbors have values around 27000 and higher. One of the disadvantages of **automap** is that there occur predicted values that are negative. Moreover, predictive confidence intervals all have approximately the same length no matter if the predicted value is very small or very high. From a qualitative point of view the copula-based model fits better to the data because it accounts for the fact that all measurements are positive and very high predicted values are typically more uncertain than low predicted values.

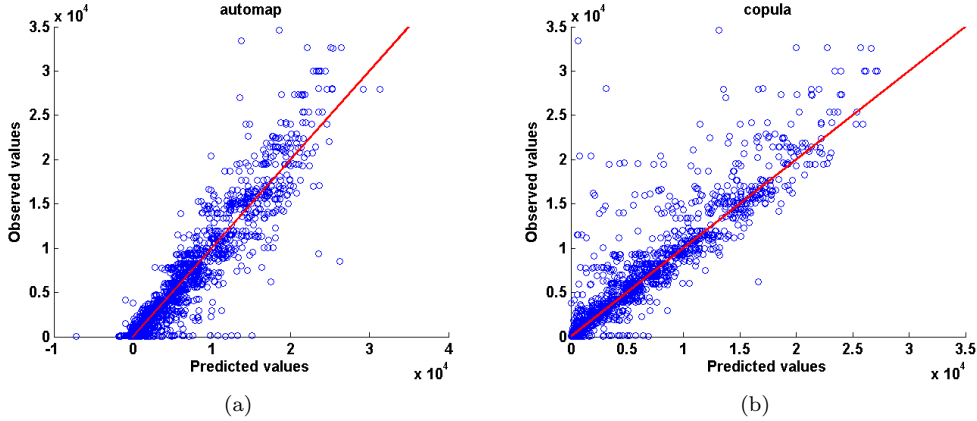


Figure 2: Scenario 6 data: Predicted values plotted against the true values for the (a) **automap** and the (b) **copula** prediction method.

5 Conclusions

Within the INTAMAP project we have developed novel statistical techniques for interpolation of non-Gaussian data based on copula functions. This methodology is integrated in the `intamap` R-library, which will be uploaded to CRAN. The program works with two copula functions, the Gaussian and the χ^2 -copula, and performs maximum likelihood parameter estimation and plug-in prediction at specified prediction locations. We have designed our code for fully automatic interpolation as is required for the WPS. However, it is also easy to use as a stand-alone tool for interactive use within R.

Bibliography

- Bardossy, A. (2006), Copula-based geostatistical models for groundwater quality parameters. *Water Resources Research*, 42, W11416
- Bardossy, A. & Li, J. (2008), Geostatistical interpolation using copulas. *Water Resources Research*, 44, W07412
- Huber, P. & Ronchetti, E. (2009), *Robust statistics*. Wiley, New York
- Kazianka, H. & Pilz, J. (2009a), Spatial interpolation using copula-based geostatistical models. In: P. Atkinson and C. Lloyd (eds.) *GeoENV VII - Geostatistics for Environmental Applications*. Springer, Berlin
- Kazianka, H. & Pilz, J. (2009b), Copula-based geostatistical modeling of continuous and discrete data including covariates. *Stochastic Environmental Research and Risk Assessment* (accepted)
- Nelsen, R. (2006), *An Introduction to Copulas*. Springer, New York
- De Oliveira, V., Kedem, B. & Short, D. (1997), Bayesian prediction of transformed Gaussian fields. *Journal of the American Statistical Association*, 92, 1422-1433
- Stöhlker, U., Dubois, G., De Jesus, J., Burbeck, S., Bleher, M. & Pebesma, E. (2009), Realtime mapping for environmental surveillance: A decision-maker's perspective. In: G. Dubois (ed.) *Proceedings StatGIS 2009*

A Help files of the spatialCopula R-library

spatialCopula-package

spatialCopula - Implements the spatial copula methodology

Description

The spatialCopula package provides functions that perform spatial modelling and spatial interpolation using copulas. The use of copulas makes it possible to overcome the Gaussian assumption often made in geostatistics.

Details

Package: spatialCopula
Type: Package
Version: 1.6-3
Date: 2009-05-20
License: What license is it under?
LazyLoad: yes

- `bayesCopula`: performs spatial prediction with given parameters
- `copulaEstimation`: estimates the model parameters given initial parameters
- `estimateParameters.copula`: estimates the model parameters for an Intamap object of type "copula" (uses `copulaEstimation`)
- `spatialPredict.copula`: performs spatial prediction for an Intamap object of type "copula"
- `intamapExampleObject`: Intamap object of class "copula" containing a simulated data set

Author(s)

Author and maintainer: Hannes Kazianka, University of Klagenfurt, hannes.kazianka@uni-klu.ac.at

References

Kazianka, H. and Pilz, J. (2009), Spatial Interpolation Using Copula-Based Geostatistical Models. GeoENV2008 - Geostatistics for Environmental Application (P. Atkinson, C. Lloyd, eds.), Springer, New York

Examples

```
data(intamapExampleObject)
## estimate parameters for the copula model
## Not run: intamapExampleObject<-estimateParameters.copula(intamapExampleObject)
## make predictions at unobserved locations
## Not run: intamapExampleObject<-spatialPredict.copula(intamapExampleObject)
```

`estimateParameters.copula`

Spatial Modelling for Intamap Object

Description

Performs copula-based spatial modeling for an Intamap object of class "copula".

Usage

```
estimateParameters.copula(object)
```

Arguments

`object` Intamap object, see description in [00Introduction](#)

Details

By default the correlation function model and the starting values for its parameters are chosen according to `autofitVariogram`. `estimateAnisotropy` decides whether to consider geometric anisotropy or not and calculates the corresponding starting values. A suitable family of marginal distributions is chosen on the basis of a Kolmogorov-Smirnov test and the starting values for their parameters are set to the maximum likelihood estimate under the independence hypothesis. For more information regarding the optimization procedure see `copulaEstimation`.

Value

Returns updated intamap object. The estimated parameters are saved in `object$copulaParams`.

Author(s)

Hannes Kazianka

References

Kazianka, H. and Pilz, J. (2009), Spatial Interpolation Using Copula-Based Geostatistical Models. GeoENV2008 - Geostatistics for Environmental Application (P. Atkinson, C. Lloyd, eds.), Springer, New York

See Also

`copulaEstimation`, `spatialPredict.copula`, `bayesCopula`

Examples

```
data(intamapExampleObject)
## estimate parameters for the copula model
## Not run: intamapExampleObject<-estimateParameters.copula(intamapExampleObject)
## make predictions at unobserved locations
## Not run: intamapExampleObject<-spatialPredict.copula(intamapExampleObject)
```

`copulaEstimation` *ML-estimation of the spatial copula model parameters*

Description

Estimates parameters of the spatial copula model using maximum likelihood.

Usage

```
copulaEstimation(obj,margin,trend,correlation,anisotropy,copula,tol=0.001)
```

Arguments

<code>obj</code>	Intamap object, see description in <code>00Introduction</code>
<code>margin</code>	list with the following elements: <ul style="list-style-type: none"><code>params</code> Starting values for the parameters of the marginal distribution (excluding trend parameters)<code>lower</code> Lower bounds for the values of the parameters of the marginal distribution (excluding trend parameters)<code>upper</code> Upper bounds for the values of the parameters of the marginal distribution (excluding trend parameters)<code>name</code> Name of the family of marginal distributions. Possible names are: "norm", "lnorm", "gev", "t" and "logis"
<code>trend</code>	list with the following elements: <ul style="list-style-type: none"><code>params</code> Starting values for the parameters of the trend model (location parameter of the marginal distribution)

lower Lower bounds for the values of the parameters of the trend model
 upper Upper bounds for the values of the parameters of the trend model
 F Design matrix.

correlation list with the following elements:

model Correlation function model. Possible models are: "Ste", "Sph",
 "Gau" and "Exp"

params Starting values for the parameters of the correlation function model

lower Lower bounds for the values of the parameters of the correlation
 function model

upper Upper bounds for the values of the parameters of the correlation
 function model

anisotropy list with the following elements:

params Starting values for the parameters of geometric anisotropy. If
 NULL, then no anisotropy is considered.

lower Lower bounds for the values of the parameters of geometric anisotropy.
 Usually $c(0, 1)$

upper Upper bounds for the values of the parameters of geometric anisotropy.
 Usually $c(\pi, \text{Inf})$

copula list with the following elements:

method Either "norm" or "chisq", depending on which spatial copula model
 is used, the Gaussian or the chi-squared copula.

params Only used in case of the chi-squared copula: the squared non-
 centrality parameter of the non-central chi-squared distribution.
 Controls how far the chi-squared copula is from the Gaussian cop-
 ula.

lower Only used in case of the chi-squared copula: the lower bound for
 the copula parameter. Usually set to 0

upper Only used in case of the chi-squared copula: the upper bound for
 the copula parameter. Usually set to **Inf**

tol Tolerance level for the optimization process.

Details

copulaEstimation performs maximum likelihood estimation of all possible param-
 eters included in the Gaussian and chi-squared spatial copula model: parameters of the
 predefined family of marginal distributions (including spatial trend or external drift),
 correlation function parameters, parameters for geometric anisotropy and parameters
 for the copula (only used for the chi-squared copula model). Due to the large number
 of variables that need to be optimized, a profile-likelihood approach is used. Although
 convergence to a global optimum is not assured, the profile-likelihood method makes
 it less likely that the optimization routine, **optim**, gets stuck in a local optimum. The
 result of **copulaEstimation** is a list containing all parameter point estimates that are
 needed for plug-in spatial prediction. It is advisable to check the output of the algorithm
 by trying different starting values for the optimization.

Value

A list with the following elements:

<code>margin</code>	Same as the input except that the list element "params" now consists of the optimized parameters of the marginal distribution function.
<code>trend</code>	Same as the input except that the list element "params" now consists of the optimized parameters of the trend model.
<code>correlation</code>	Same as the input except that the list element "params" now consists of the optimized parameters of the correlation function model.
<code>anisotropy</code>	Same as the input except that the list element "params" now consists of the optimized parameters of geometric anisotropy.
<code>copula</code>	Same as the input except that the list element "params" now consists of the optimized copula parameters.

Author(s)

Hannes Kazianka

References

Kazianka, H. and Pilz, J. (2009), Spatial Interpolation Using Copula-Based Geostatistical Models. GeoENV2008 - Geostatistics for Environmental Application (P. Atkinson, C. Lloyd, eds.), Springer, New York

See Also

`bayesCopula`, `spatialPredict.copula`, `estimateParameters.copula`

Examples

```
data(intamapExampleObject)
## estimate parameters for the copula model
## Not run:
copula<-list(method="norm")
anisotropy<-list(lower=c(0,1),upper=c(pi,Inf),params=c(pi/3,2))
correlation<-list(model="Ste",lower=c(0.01,0.01,0.01),upper=c(0.99,Inf,20),params=c(0.05,4,3))
margin<-list(name="gev",lower=c(0.01,-Inf),upper=c(Inf,Inf),params=c(30,0.5))
trend<-list(F=as.matrix(rep(1,196)),lower=-Inf,upper=Inf,params=40)
estimates<-copulaEstimation(intamapExampleObject,margin,trend,correlation,anisotropy,copula)
## End(Not run)
## make predictions at unobserved locations
## Not run: predictions<-bayesCopula(intamapExampleObject,estimates,search=25,
calc=list(mean=TRUE,variance=TRUE,excprob=40,quantile=0.95))
```

`spatialPredict.copula`

Spatial Interpolation for Intamap Object

Description

Performs copula-based spatial prediction for an Intamap object of the class "copula".

Usage

```
spatialPredict.copula(object)
```

Arguments

`object` Intamap object, see description in `00Introduction`

Details

`spatialPredict.copula` is just a wrapper for `bayesCopula`. It is possible to calculate predictive mean, variance and quantiles as well as exceedance probabilities above certain thresholds. `object$outputWhat` specifies which predictive statistics are sought. `object$params$nmax` defines how many observed locations should be used for local prediction (maximum is 50).

Value

Returns updated intamap object. The estimated parameters are saved in `object$predictions`.

Author(s)

Hannes Kazianka

References

Kazianka, H. and Pilz, J. (2009), Spatial Interpolation Using Copula-Based Geostatistical Models. *GeoENV2008 - Geostatistics for Environmental Application* (P. Atkinson, C. Lloyd, eds.), Springer, New York

See Also

`copulaEstimation`, `bayesCopula`, `estimateParameters.copula`

Examples

```
data(intamapExampleObject)
## estimate parameters for the copula model
## Not run: intamapExampleObject<-estimateParameters.copula(intamapExampleObject)
## make predictions at unobserved locations
## Not run: intamapExampleObject<-spatialPredict.copula(intamapExampleObject)
```

bayesCopula	<i>Performs spatial interpolation using copulas</i>
-------------	---

Description

Calculates predictive mean, predictive variance, predictive quantiles and exceedance probabilities for certain thresholds in the spatial copula model.

Usage

```
bayesCopula(obj, estimates, search=10, calc=list(mean=TRUE, variance=TRUE), testMean=FALSE)
```

Arguments

<code>obj</code>	Intamap object including observations and predictionLocations, see <code>00Introduction</code>
<code>estimates</code>	List of estimated parameters (typically obtained by calling <code>copulaEstimation</code>)
<code>search</code>	Local prediction: number of observed locations considered for prediction at each unknown point
<code>calc</code>	List of what prediction type is required: <code>mean = TRUE</code> TRUE if the predictive mean should be calculated, FALSE otherwise <code>variance = TRUE</code> TRUE if the predictive variance should be calculated, FALSE otherwise <code>quantiles = NULL</code> Vector of desired predictive quantiles, e.g. 0.95 or 0.05 <code>excprob = NULL</code> Vector of thresholds, where the probability of exceeding this threshold is desired
<code>testMean</code>	Whether or not the mean estimates should be tested if they are reasonable.

Details

`bayesCopula` is used for plug-in prediction at unknown spatial locations. The name of the function is somewhat misleading since no Bayesian approach is implemented so far. It is possible to calculate numerically the predictive mean and variance for both the Gaussian and the chi-square spatial copula model. Exceedance probabilities and predictive quantiles are only supported for the Gaussian copula model. Note that

it may occur that the predictive distribution has no finite moments. In this case, a possible predictor is the median of the predictive distribution. If `testMean=TRUE` and the predictive means have no reasonable values, the median is automatically calculated and a warning is produced.

Value

List with the following elements:

<code>mean</code>	Mean of the predictive distribution. NULL if not calculated.
<code>variance</code>	Variance of the predictive distribution. NULL if not calculated.
<code>quantiles</code>	Quantiles of the predictive distribution NULL if not calculated.
<code>excprob</code>	Probabilities for the predictive distribution to exceed predefined thresholds. NULL if not calculated.

Author(s)

Hannes Kazianka

References

Kazianka, H. and Pilz, J. (2009), Spatial Interpolation Using Copula-Based Geostatistical Models. GeoENV2008 - Geostatistics for Environmental Application (P. Atkinson, C. Lloyd, eds.), Springer, New York

See Also

`copulaEstimation`, `spatialPredict.copula`, `estimateCopula`

Examples

```
data(intamapExampleObject)
## estimate parameters for the copula model
## Not run:
copula<-list(method="norm")
anisotropy<-list(lower=c(0,1),upper=c(pi,Inf),params=c(pi/3,2))
correlation<-list(model="Ste",lower=c(0.01,0.01,0.01),upper=c(0.99,Inf,20),params=c(0.05,4,3))
margin<-list(name="gev",lower=c(0.01,-Inf),upper=c(Inf,Inf),params=c(30,0.5))
trend<-list(F=as.matrix(rep(1,196)),lower=-Inf,upper=Inf,params=40)
estimates<-copulaEstimation(intamapExampleObject,margin,trend,correlation,anisotropy,copula)
## End(Not run)
## make predictions at unobserved locations
## Not run: predictions<-bayesCopula(intamapExampleObject,estimates,search=25,
calc=list(mean=TRUE,variance=TRUE,excprob=40,quantile=0.95))
```

`intamapExampleObject` *Simulated Intamap Object*

Description

Intamap object of class "copula" containing a simulated data set with 196 spatial locations.

Usage

```
data(intamapExampleObject)
```

Details

The data set is a realization of a random field generated using a Gaussian copula and generalized extreme value distributed margins (location=40,shape=0.5, scale=30). The correlation function is Matern (Stein's representation) with range=4, kappa=3 and nugget=0.05. Furthermore, there is geometric anisotropy with direction=pi/3 and ratio=2.

See Also

```
spatialPredict.copula, estimateParameters.copula
```

Examples

```
data(intamapExampleObject)
## estimate parameters for the copula model
## Not run: intamapExampleObject<-estimateParameters.copula(intamapExampleObject)
## make predictions at unobserved locations
## Not run: intamapExampleObject<-spatialPredict.copula(intamapExampleObject)
```