Smooth Spatial Maximum Likelihood GEV Fitting with gevreg

Harald Schellander
ZAMG Innsbruck

Abstract

The **gevreg** package provides functions for maximum likelihood estimation of smooth spatial regression models for the Generalized Extreme Value Distribution GEV. Models for the location, scale and shape parameter of the GEV can have different regressors. Suitable standard methods to compute predictions are provided as well. The model and its R implementation is introduced and illustrated by usinf snow depth data for Austria.

Keywords: regression, GEV, smooth spatial, R.

1. Introduction

A spatial representation of meteorological extremes such as snow depth is of crucial importance for numerous purposes such as the planning and construction of buildings, for avalanche simulation (Rudolf-Miklau and Sauermoser 2011) or in general risk assessment.

A more or less simple interpolation of GEV parameters to space has some disadvantages, as e.g. Blanchet and Lehning (2010) showed for extreme snow depths in Switzerland. As an improvement they suggested a direct estimation of a spatially smooth generalized extreme value (GEV) distribution, called *smooth spatial modeling*. With smooth modeling the GEV parameters are modeled as smooth functions of spatial covariates. Spatially varying marginal distributions are achieved by maximizing the sum of the log-likelihood function over all stations. Compared to several interpolation methods, smooth modeling for swiss snow depth led to more accurate marginal distributions, especially in data sparse regions. The key feature of smooth modeling, permitting to approximate the likelihood as a sum of GEV likelihoods at the stations, is the simplifying assumption that annual snow depth maxima are approximately independent in space and time. Nevertheless, smooth modeling does not provide any spatial dependence of extremes.

As a way to account for spatial dependence of extremes, *max-stable processes* as an extension of multivariate extreme value theory to infinite dimensions can be used (de Haan 1984). With max-stable processes, the margins and their spatial dependency can be modeled simultaneously but independently. The **SpatialExtremes** (Ribatet 2017) package provides functions for statistical modelling of spatial extremes using max-stable processes, copula or Bayesian hierarchical models. In addition, the **hkevp** (Sebille 2016) package provides several procedures around the HKEVP model of Reich and Shaby (2012) and the Latent Variable Model of Davison, Padoan, and Ribatet (2012). However, no package for easy use of the smooth modeling approach exists.

Therefore, the **gevreg** package provides a function to fit a smooth spatial regression model to observations. It has a convenient interface to estimate the model with maximum likelihood and provides several methods for analysis and prediction. The outline of the paper is as follows. Section 2 describes the idea of smooth spatial (extreme value) modeling, and Section 3 presents its R implementation. Section 4 illustrates the package functions with Austrian snow depth data and finally Section 5 summarizes the paper.

2. Smooth Spatial Modeling of Extremes

The annual maximum of a variable can be interpreted as a time-space stochastic process $\{S_x^{(t)}\}$, where t denotes the corresponding year and $x \in \mathcal{A}$ the location ,e.g. in Austria. However, we assume that the distribution of $S_x^{(t)}$ does not depend on the time t and therefore for each of the GEV parameters, we consider a linear model, i. e. a model of the form

$$\eta(x) = \alpha_0 + \sum_{k=1}^{m} \alpha_k y_k(x) + f\left(y_{m+1}(x), \ldots, y_n(x)\right) \tag{1}$$

at location x, where η denotes one of the GEV parameters, $y_1, \ldots, y_n$ are the considered covariates as functions of the location, $\alpha_0, \ldots, \alpha_m \in \mathbb{R}$ are the coefficients of the linear part and f is a P-spline with 2 knots, evenly distributed across the spatial domain. For $k = 1, \ldots, K$ the k-th station is given by the location x_k and therefore, we have a realization $s_{x_k}^{(1)}, \ldots, s_{x_k}^{(N)}$ of the random sample $S_{x_k}^{(1)}, \ldots, S_{x_k}^{(N)}$ given as measurements. Note that $S_{x_k} \sim \mathrm{GEV}(\mu_{x_k}, \sigma_{x_k}, \xi_{x_k})$ and $\mu(x_k), \sigma(x_k), \xi(x_k)$ are the GEV parameters given by the linear models in (1). By $\ell_k\left(\mu(x_k), \sigma(x_k), \xi(x_k)\right)$ we denote the log-likelihood function at the k-th station corresponding to (1). With the assumption of spatially independent stations, the log-likelihood function then reads as

$$l = \sum_{k=1}^{K} \ell_k\left(\mu(x_k), \sigma(x_k), \xi(x_k)\right),$$

where l only depends on the coefficients of the linear models for the GEV parameters, cf. (1). This approach was called *smooth modeling* by Blanchet and Lehning (2010).

The advantage of maximizing the sum of the log-likelihood functions at the stations compared to maximizing the log-likelihood function at each station lies in the following fact: A good fit at a single station leading to worse fits at several other stations will be penalized. As a consequence, the stations become intertwined in terms of the fitting. As the smooth model does not provide any spatial dependence, it is generally assumed to be less suited to spatially model extremes, compared to other approaches as fitting a max-stable process.

3. R implementation

As other R regression packages the main function `gvereg()` uses a formula-based interface and returns an object of class `gevreg`:

```
gevreg <- function (formula, data, subset, na.action,
                    model = TRUE, y = TRUE, x = TRUE, z = FALSE, v = FALSE,
                    gev_params, control = gevreg_control(...),
                    ...)
```

The `formula` is actually of class **Formula** (Zeileis and Croissant 2010) and can have three parts separated by '`|`', specifying potentially different sets of regressors x_i, z_i and v_i for the location, scale and shape submodels, respectively. This function `gevreg()` takes this formula and fits the given (linear) submodels to the `gev_params` by using the R function `optimx()` to maximise the likelihood. The GEV parameters `gev_params` to which the submodels are fitted, can either be supplied externally, in which case, they must be an R `data.frame` with columns "loc", "scale", and "shape" and one row per observation. If no `gev_params` are supplied, they will be computed internally from the `data` using the function `gevmle` from the package **SpatialExtremes**. In any case, they are returned with the fitted object.

The maximum likelihood estimation is performed with `optimx()` using control options specified in `gevreg_control()`. By default optimization is carried out with method "nlminb". If option `grad = FALSE`, "Nelder-Mead" is used instead. If no starting values are supplied with `start`, the coefficients from `glm()` are used as starting values for the location and shape parameter. For the scale parameter model a log-link function is applied to the offset of the coefficients of the `glm()` output. The absolute value is taken for the rest of the coefficients.

The actual fitting function is `gevreg_fit()`. As mentioned in Section 2, it maximises the sum of the log-likelihoods at each station. After a proper rescaling of the covariates, the actual fitting is achieved in two steps: if the "standard" negative log-likelihood returns `Inf`, a "squared" log-likelihood function is tried to be maximised. `gevreg_fit()` has a matrix interface and returns an unclassed list.

The object returned by `gevreg()` is a list, similar to that of the "glm" objects. In addition, model frame, response and regressor matrices are returned if specified in the call to `gevreg()`.

Note that in the current implementation `na.action = NULL` has to set explicitly. This will be corrected in a future version.

4. Example

In this Section Austrian sow depth data at 36 observation sites are used to fit a smooth spatial GEV model with **gevreg**. Then the S3 function `predict()` is applied to the fitted object to (1) generate "location" parameters at a grid, and (2) compute 50-year return levels of snow depth on that grid.

First, the **gevreg** package is loaded together with included Austrian snow depth data. In addition the plot package **ggplot2** is loaded.

```
R> library("gevreg")
R> library("ggplot2")
R> data("SnowAustria")
R> data("SnowAustriaGEV")
R> data("SnowAustriaMap")
R>
R>
```

The dataset `SnowAustria` contains a list with three components. First, `SnowAustria` holds Austrian snow depth data as well as coordinates (longitude, latitude, altitude) in a data frame. The first column (the name "station" is mandatory) is a factor variable containg the levels 1 to 36. Column 2 (hs) contains snow depth maxima for 68 years per station, including NA's. Columns 3 to 5 (lon, lat, alt) are coordinates within the domain of Austria. Second, `SnowAustriaGEV` is a data frame with GEV parameters valid at the stations. The three columns "loc", "scale" and "shape" correspond to the location, scale and shape parameters locally estimated at the stations using function `gevmle` of package **SpatialExtremes**. And third, a grid for plotting a map of extremes for the state of Tyrol, named `SnowAustriaMap`. Figure 1 shows the geographical position of the stations, revealing a good variance over Austria.

```
R> ggplot(SnowAustria,aes(lon,lat,color=alt)) +
+    geom_point(size = 4) +
+    annotate("label",
+              x = c(11.3927,16.373),
+              y = c(47.2672,48.203),
+              label = c("Innsbruck","Wien"))
```

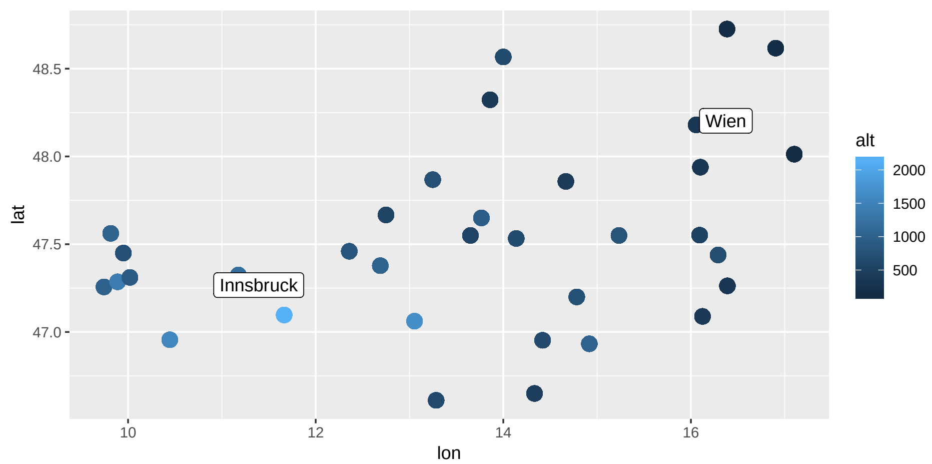

Figure 1: Geographical distribution of stations used in dataset `SnowAustria`.

A quick check of the snow depth values indicate indeed that they might exhibit a heavy tail, suited to be modeled with the GEV distribution (Figure 2):

```
R> ggplot(SnowAustria,aes(hs)) +
+    geom_histogram(bins = 20) +
+    labs(x = "height of snow [cm]")
```

Exploring scatterplots of the GEV parameters against geographical coordinates indicates some relationships, helpful for setting up the submodels for the GEV parameters (Figure 3):

```
R> df <- cbind(SnowAustriaGEV,data.frame(lon = unique(SnowAustria$lon), alt = unique(SnowA
R> plot(df)
```

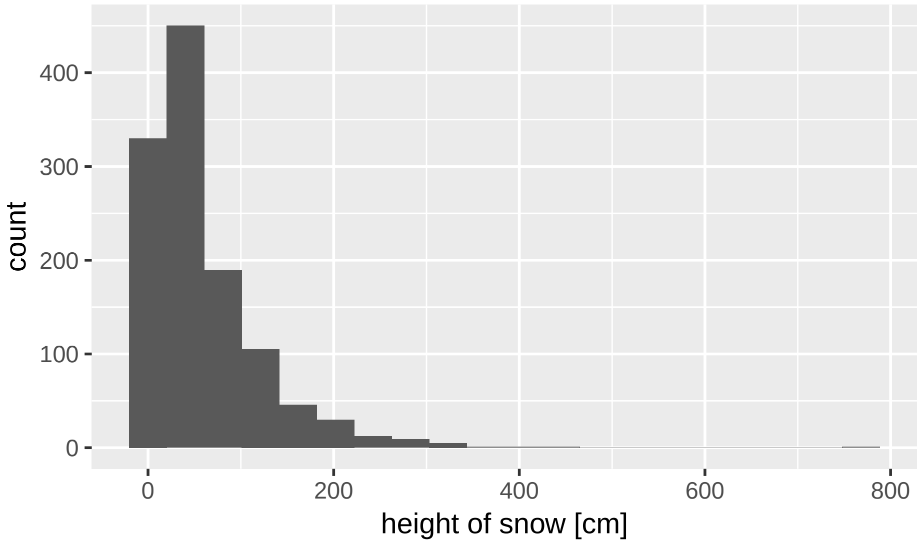

Figure 2: The histogram of snowdepth values in dataset `SnowAustria` shows a heavy tail.

From this we can set up a call to `gevreg()`. First, the location parameter is added as column to the data.frame SnowAustria as covariate:

```
R> SnowAustria$loc <- as.numeric(rep(SnowAustriaGEV$loc,
+                                    each=nrow(SnowAustria)/
+                                      length(levels(SnowAustria$station))))
```

Then the function call is set up, using “lon” and “alt” as covariates for the location parameter, “lat”, “alt” and the location parameter “loc” as covariate for the scale parameter, and the longitude “lon” as covariate for the shape parameter:

```
R> m0 <- gevreg(formula = hs ~ lon + alt | lat + alt + loc | lon,
+                 data = SnowAustria, gev_params = SnowAustriaGEV, na.action = NULL)
```

The fitted model summarizes as follows:

```
R> print(m0)
```

```
Smooth Spatial GEV Fitting

Estimator: Maximum Likelihood
AIC: 11232.202150
BIC: 11284.429390

Call:
gevreg(formula = hs ~ lon + alt | lat + alt + loc | lon, data = SnowAustria,
    na.action = NULL, gev_params = SnowAustriaGEV)
```

```
R> df <- cbind(SnowAustriaGEV,data.frame(lon = unique(SnowAustria$lon),
+                                        alt = unique(SnowAustria$alt)))
R> plot(df)
```

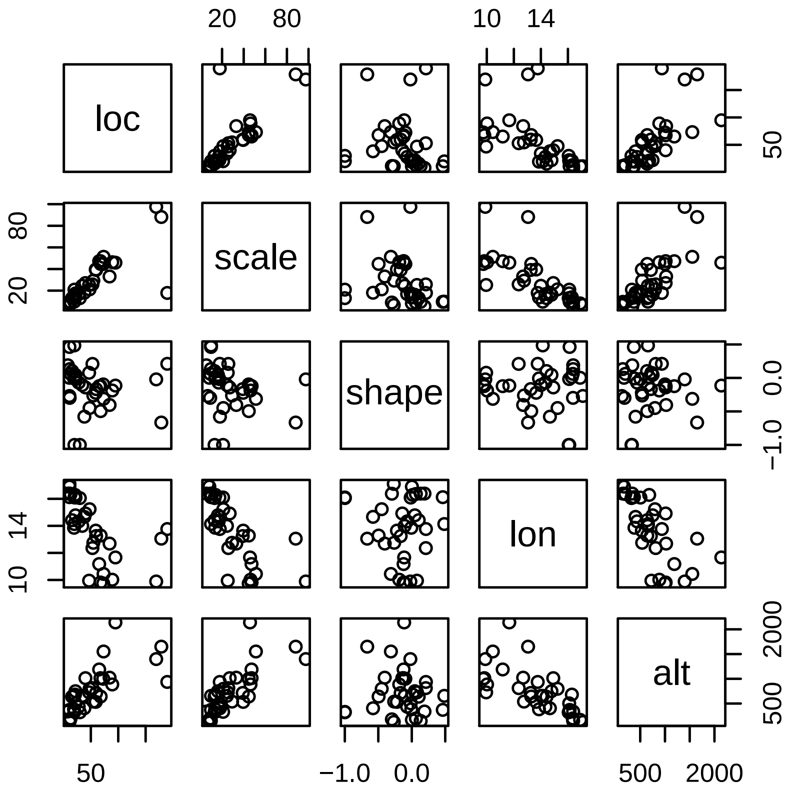

Figure 3: Relations between the GEV parameters and covariables lon and alt in the dataset `SnowAustria`, that can be used to set up the marginal models.

```
Convergence: successful, method: nlminb

Coefficients (location model):
(Intercept)          lon          alt
    91.1051      -5.0606       0.0179

Coefficients (scale model with log link):
(Intercept)          lat          alt          loc
   -6.04548      0.17619      0.01104      0.44277

Coefficients (shape model):
(Intercept)          lon
 -0.0465964   -0.0005062

Log-likelihood: -5607 on 9 Df
```

The full model from above can be compared with a simpler model. Taking e.g. a constant location parameter over the whole domain of Austria:

```
R> m1 <- gevreg(formula = hs ~ 1 | lat + alt + loc | lon,
+                data = SnowAustria, gev_params = SnowAustriaGEV, na.action = NULL)
```

But as expected, this leads to a worse fitting performance, as can be seen by comparing the AIC and the maximised likelihood:

```
R> cbind(AIC(m0,m1),loglik=c(logLik(m0),logLik(m1)))
```

```
   df      AIC    loglik
m0  9 11232.20 -5607.101
m1  7 11586.99 -5786.496
```

By applying the S3 function `predict` to the fitted `gevreg` object m0, return values on a grid can be computed. The grid has to have an extra column with the covariate "loc". To get a map of extreme snow depths with a return peripod of 50 years on the grid provided by `SnowAustria`, one could use the following R-code, which results in Figure 4:

```
R> SnowAustriaMap$loc <- predict(m0, type = "location",
+                               newdata = SnowAustriaMap, at=1-1/50)
```

Results can then be plotted with ggplot:

```
R> library("ggplot2")
R> library("colorspace")
R> values <- c(0,150,200,300,400)
R> colors <- diverge_hcl(length(values),
+                        h = c(260, 0),
```

```
+                           c = 100,
+                           l = c(30, 90),
+                           power = 0.7)
R> rl <- predict(m0, type = "quantile", newdata = SnowAustriaMap, at=1-1/50)
R> rl.fact <- cut(rl,values,values[-1])
R> ggplot(data.frame(SnowAustriaMap,rl.fact),aes(lon,lat,color=rl.fact,fill=rl.fact)) +
+    geom_point(shape=22, size=2.75) +
+    scale_color_manual(values = colors[1:length(values)],
+                       limits = values,
+                       breaks = rev(values[-1])) +
+    scale_fill_manual(values = colors[1:length(values)],
+                      limits = values,
+                      breaks = rev(values[-1])) +
+    guides(fill = guide_legend("50-year Snow Depth [cm]"),
+           title.hjust = 0.5,
+           color = FALSE) +
+    annotate("point",x=11.393,y=47.267,shape=4,size=5,stroke=0.3) +
+    annotate("text",x=11.4,y=47.267,label="Innsbruck",size=5,hjust=-0.2)
```

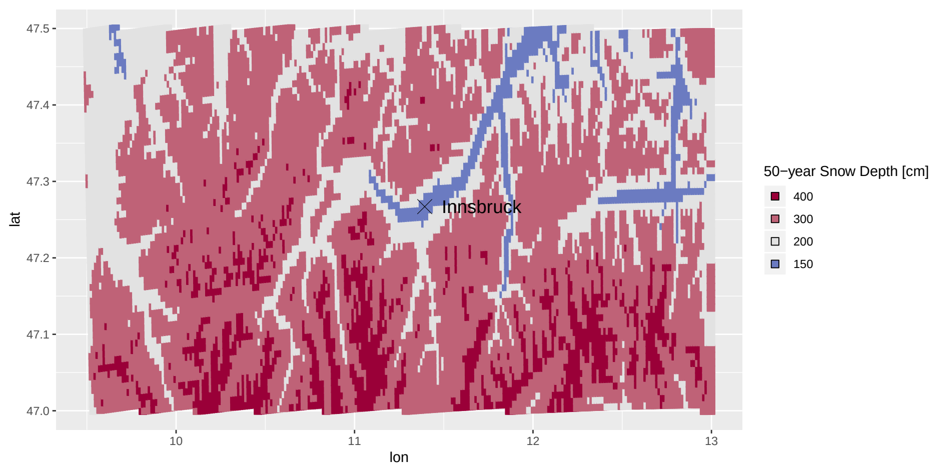

Figure 4: 50-year snow depth return levels computed with a smooth GEV model. A 50-year return level of roughly 130 cm is modeled for the Inn valley. The high altitudes show larger extremes.

5. Summary

The spatial modeling of extremes has a broad spectrum of application e.g. in Meteorology and Climatology. While there exist other approaches like max-stable processes, which account for the spatial extremal dependence as well, smooth spatial modeling can be a viable alternative. If ones focus lies more on exact margins, than on risk modeling, the package `gevreg` can easily be used to compute spatial quantiles of GEV distributed extremes. The use of the package was illustrated with a return level map of snow depths of the region of Tyrol.

6. Acknowledgements

The core of the code of package **gevreg** is based on the Masterthesis of Simon Gstöhl, which he did during a project at ZAMG (Gstöhl 2017).

References

Blanchet J, Lehning M (2010). "Mapping snow depth return levels: smooth spatial modeling versus station interpolation." *Hydrol. Earth Syst. Sci*, **14**(12), 2527–2544. doi:10.5194/hess-14-2527-2010. URL http://www.hydrol-earth-syst-sci.net/14/2527/2010/.

Davison AC, Padoan SA, Ribatet M (2012). "Statistical Modeling of Spatial Extremes." *Statistical Science*, **27**(2), 161–186. doi:10.1214/11-STS376. URL http://dx.doi.org/10.1214/11-STS376.

de Haan L (1984). "A Spectral Representation for Max-stable Processes." *The Annals of Probability*, **12**(4), 1194–1204. ISSN 0091-1798. doi:10.1214/aop/1176993148. URL http://dx.doi.org/10.1214/aop/1176993148http://www.jstor.org/stable/2243357.

Gstöhl S (2017). *Spatial modeling of extreme snow depth and snow water equivalent*. Masterthesis, University of Innsbruck.

Reich BJ, Shaby BA (2012). "A hierarchical max-stable spatial model for extreme precipitation." *Annals of Applied Statistics*, **6**(4), 1430–1451. doi:10.1214/12-AOAS591. URL http://dx.doi.org/10.1214/12-AOAS591.

Ribatet M (2017). *SpatialExtremes: Modelling Spatial Extremes*. R package version 2.0-5, URL https://CRAN.R-project.org/package=SpatialExtremes.

Rudolf-Miklau F, Sauermoser S (2011). *Handbuch Technischer Lawinenschutz*. Ernst & Sohn. A Wiley Company.

Sebille Q (2016). *hkevp: Spatial Extreme Value Analysis with the Hierarchical Model of Reich and Shaby (2012)*. R package version 1.1.4, URL https://CRAN.R-project.org/package=hkevp.

Zeileis A, Croissant Y (2010). "Extended Model Formulas in R: Multiple Parts and Multiple Responses." *Journal of Statistical Software*, **34**(1), 1–13. URL http://www.jstatsoft.org/v34/i01/.

Affiliation:

Harald Schellander
ZAMG - Zentralanstalt für Meteorologie und Geodynamik
6020 Innsbruck, Austria
E-mail: Harald.Schellander@zamg.ac.at