SpatialGEV: Fast Bayesian inference for spatial extreme value models in R

18 October 2021

Summary

Extreme weather phenomena such as floods and hurricanes are of great concern due to their potential to cause extensive damage. To develop more reliable damage prevention protocols, statistical models are often used to infer the chance of observing an extreme weather event at a given location (Coles and Casson 1998; Cooley, Nychka, and Naveau 2007; Sang and Gelfand 2010). The R package `SpatialGEV` is a fast and convenient tool for analyzing spatial extreme values using the hierarchical GEV-GP model in a Bayesian framework. In this model, the marginal behavior of the extremes is described using the generalized extreme value (GEV) distribution, whereas the spatial dependence is captured by modelling the GEV parameters as spatially varying random effects following Gaussian processes (GP). Users are provided with a streamlined way to build and fit their models in R, which are compiled in C++ under the hood. The complexity of the model can flexibly vary depending on how many GEV parameters are considered random. For downstream analysis, this package offers methods for making Bayesian inferences about the model parameters and forecasting extreme events.

Statement of need

The GEV-GP model has important applications in meteorological studies. With extreme rainfall as an example, it is often of interest for meteorologists to estimate its $p\%$ return level $z_p(\boldsymbol{x})$ at a given location $\boldsymbol{x}$, which is the value above which precipitation levels at location $\boldsymbol{x}$ occur with probability p. Simply put, $z_p(\boldsymbol{x})$ is the $p \times 100\%$ upper quantile of the GEV distribution at location x. When p is chosen to be a small value, $z_p(\boldsymbol{x})$ indicates how extreme the precipitation level might be at location $\boldsymbol{x}$. Thus, knowledge about the posterior distribution $p(z_p(\boldsymbol{x}) \mid \boldsymbol{y})$ is useful for forecasting extreme weather events. In practice, Markov Chain Monte Carlo (MCMC) methods are often used to sample from the posterior distribution (e.g., Cooley, Nychka, and Naveau 2007; Schliep et al. 2010; Dyrrdal et al. 2015). However, they can be extremely time-consuming when the number of locations is large, taking hours or even days before convergence. `SpatialGEV` package therefore implements Bayesian inference based on the Laplace approximation as an alternative to MCMC, making large-scale spatial analyses orders of magnitude faster while achieving the same accuracy as MCMC. The Laplace approximation is carried out using the R/C++ package `TMB` (Kristensen et al. 2016). Details of the inference method can be found in Chen, Ramezan, and Lysy (2021).

Statement of field

The R pacakge `SpatialExtremes` (Ribatet, Singleton, and R Core team 2020) is one of the most popular software for fitting spatial extreme value models, which is handled through an efficient Gibbs sampler. The language Stan and its R interface `RStan` (Stan Development Team 2020) provides off-the-shelf implementations for Hamiltonian Monte Carlo and its variants (Neal 2011; Hoffman and Gelman 2014), which are considered state-of-the-art MCMC algorithms and often used for fitting hierarchical spatial models. A well-known MCMC alternative is the `R-INLA` package (Lindgren and Rue 2015) which implements the integrated nested Laplace approximation (INLA) approach. `R-INLA` has been widely used to model spatial data, but its current implementation only allows the GEV location parameters to vary spatially. Chen, Ramezan, and Lysy (2021) compares the accuracy and speed of `SpatialGEV` to `RStan` and `R-INLA` in the case of fitting GEV-GP models.

Example

Model fitting

The main functions of the `SpatialGEV` package are `spatialGEV_fit()`, `spatialGEV_sample()`, and `spatialGEV_predict()`. This example shows how to apply these functions to analyze a simulated dataset using the GEV-GP model. We simulate 1 observation per locations on a 20×20 regular lattice on $[0, 10] \times [0, 10] \subset \mathbb{R}^2$, such that there are $n = 400$ observations in total. The GEV location and scale parameters $a(\boldsymbol{x})$ and $b(\boldsymbol{x})$ are simulated from the log density surface of bivariate normal (mixture), as illustrated in Figure 1. The GEV shape parameter is set to be $s = \log(-2)$, constant across locations.

```
# Load package and data
require(SpatialGEV)
data("simulatedData")
```

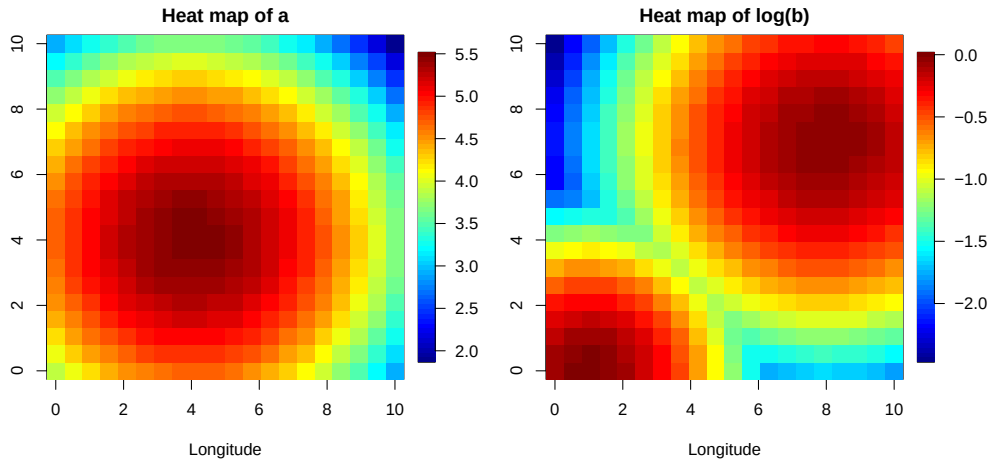

Figure 1: The simulated GEV location parameters $a(\boldsymbol{x}_i)$ and log-transformed locations parameters $\log(b)(\boldsymbol{x}_i)$ plotted on regular lattices.

The GEV-GP model is fitted by calling `spatialGEV_fit()`. By specifying `random="ab"`, both the location parameter a and scale parameter b are considered spatial random effects. Initial parameter values are passed to `init.param`. `reparam.s="positive"` means we constrain the shape parameter s to be positive. The posterior mean estimates of the spatial random effects can be accessed from `mod_fit$report$par.random`, whereas the fixed effect can be obtained from `mod_fit$report$par.fixed`.

```
mod_fit <- spatialGEV_fit(y = y, X = locs, random = "ab",
                          init.param = list(a =a, log_b = logb, s=logs,
                                            log_sigma_a = 1, log_ell_a = 5,
                                            log_sigma_b = 1, log_ell_b = 5),
                          reparam.s = "positive", silent = TRUE)
```

Sampling from the joint posterior

Now we show how to sample 5000 times from the joint posterior distribution of the GEV parameters using the function `spatialGEV_sample()`. Only three arguments need to be passed to this function: `model` takes in the list output by `spatialGEV_fit()`, `n_draw` is the number of samples to draw from the posterior distribution, and `observation` indicates whether to draw from the posterior predictive distribution of the data at the observed location. The samples are used to calculate the posterior mean estimates of the 10% return level $z_{10}(\boldsymbol{x})$ at each location, which are plotted against their true values in Figure 2.

```
require(evd)
```

```
q <- 0.1

# Sample from the joint posterior distribution of all model parameters
n_sim <- 5000
set.seed(123)
all_draws <- spatialGEV_sample((model=mod_fit, n_draw=n_sim, observation = FALSE)
all_draws <- all_draws$parameter_draws

# Calculate the posterior mean of the return levels from the samples
q_means <- rep(NA, n_loc)
s_vec <- exp(all_draws[, (2*n_loc+1)])
for (i in 1:n_loc){
  a_vec <- all_draws[ ,i]
  b_vec <- exp(all_draws[ ,i+n_loc])
  q_means[i] <- mean(apply(cbind(a_vec, b_vec, s_vec), 1,
                           function(x) evd::qgev(1-q, x[1], x[2], x[3])))
}
```

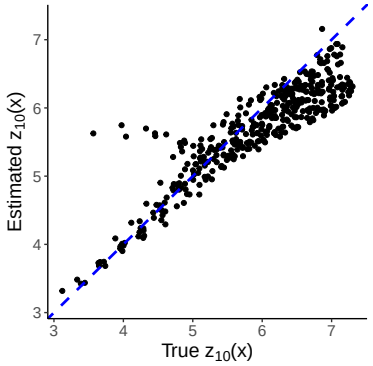

Figure 2: Posterior mean estimates of the 10$ return level $z_{10}(\boldsymbol{x})$ plotted against the true values at different locations.

Prediction at new locations

Next, we show how to make predictions of the values of the extreme event at test locations. First, we divide the simulated dataset into training and testing sets, and fit the model to the training dataset. We can simulate from the posterior predictive distribution of observations at the test locations using the `spatialGEV_predict()` function, which requires the fitted model to the training data passed to `model`, a matrix of the coordinates of the test locations passed to `X_new`, a matrix of the coordinates of the observed locations passed to `X_obs`, and the number of simulation draws passed to `n_draw`. Figure 3 plots the 95% posterior predictive intervals at the 20 test locations along with the true observed values as superimposed circles.

```
# Divide the dataset into training and testing
n_test <- 20
test_ind <- sample(1:400, n_test) # indices of the test locations
locs_test <- locs[test_ind,] # coordinates of the test locations
y_test <- y[test_ind] # observations at the test locations
locs_train <- locs[-test_ind,] # coordinates of the training (observed) locations
y_train <- y[-test_ind] # observations at the training locations

# Fit the GEV-GP model to the training set
train_fit <- spatialGEV_fit(y = y_train, X = locs_train, random = "ab",
                            init.param = list(a = a[-test_ind], log_b = logb[-test_ind], s=logs,
```

```
                                    log_sigma_a = 1, log_ell_a = 5,
                                    log_sigma_b = 1, log_ell_b = 5),
                    reparam.s = "positive", silent = TRUE)

# Make predictions at the testing locations
pred <- spatialGEV_predict(model=train_fit, X_new=locs_test, X_obs=locs_train, n_draw=5000)
```

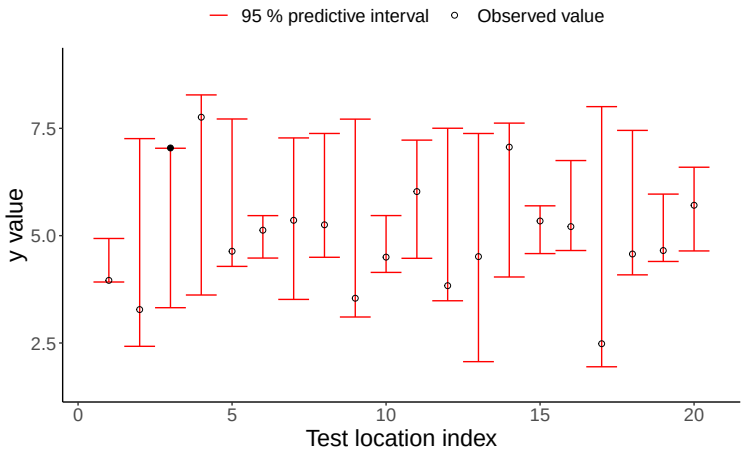

Figure 3: 95% posterior predictive intervals (PI) at test locations. Each circle corresponds to the true observation at that test location, with hollow ones indicating that they are inside the 95% PI, and solid ones indicating that they are outside of the 95% PI.

References

Chen, M., R. Ramezan, and M. Lysy. 2021. "Fast Approximate Inference for Spatial Extreme Value Models." http://arxiv.org/abs/2110.07051.

Coles, S. G., and E. Casson. 1998. "Extreme Value Modelling of Hurricane Wind Speeds." *Structural Safety* 20: 283–96.

Cooley, D., D. Nychka, and P. Naveau. 2007. "Bayesian Spatial Modeling of Extreme Precipitation Return Levels." *Journal of the American Statistical Association* 102: 824–40.

Dyrrdal, A. V., A. Lenkoski, T. L. Thorarinsdottir, and F. Stordal. 2015. "Bayesian Hierarchical Modeling of Extreme Hourly Precipitation in Norway." *Environmetrics* 26: 89–106.

Hoffman, M. D., and A. Gelman. 2014. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." *Journal of Machine Learning Research* 15: 1593–623.

Kristensen, K., A. Nielsen, C. W. Berg, H. Skaug, and B. M Bell. 2016. "TMB: Automatic Differentiation and Laplace Approximation." *Journal of Statistical Software* 70 (5): 1–21.

Lindgren, F. K., and H. Rue. 2015. "Bayesian Spatial Modelling with R-INLA." *Journal of Statistical Software* 63: 1–25.

Neal, R. M. 2011. "MCMC Using Hamiltonian Dynamics." In *The Handbook of Markov Chain Monte Carlo*. Chapman & Hall / CRC Press.

Ribatet, M., R. Singleton, and R Core team. 2020. "SpatialExtremes: Modelling Spatial Extremes." https://CRAN.R-project.org/package=SpatialExtremes.

Sang, H., and A. E. Gelfand. 2010. "Continuous Spatial Process Models for Spatial Extreme Values." *Journal of Agricultural, Biological, and Environmental Statistics* 15: 49–56.

Schliep, E. M., D. Cooley, S. R. Sain, and J. A. Hoeting. 2010. "A Comparison Study of Extreme Precipitation from Six Different Regional Climate Models via Spatial Hierarchical Modeling." *Extremes* 13: 219–39.

Stan Development Team. 2020. "RStan: The R Interface to Stan." http://mc-stan.org/.