Applying Max-and-Smooth to the UKCP data

Brynjólfur Gauti Guðrúnar Jónsson

2024-10-09

Introduction

This document describes the implementation of the Max-and-Smooth algorithm for fast approximate Bayesian inference in spatial extreme value analysis of climate projections provided by the UKCP. The algorithm is specifically applied to Generalized Extreme Value (GEV) distributions and is implemented in C++ with R interfaces using Rcpp and RcppEigen.

Package Overview

The `maxandsmooth` R package provides tools for fast approximate Bayesian inference for spatial GEV models. The core of the package is implemented in C++ for efficiency, with R wrappers for ease of use.

Key features of the package include:

- Implementation of the Max-and-Smooth algorithm for GEV distributions
- Efficient C++ code using Eigen for linear algebra operations
- Spatial modeling of GEV parameters using Intrinsic Conditional Autoregressive (ICAR) priors
- R interface for easy integration into existing extreme value analysis workflows

Algorithm Description

The Max-and-Smooth algorithm, as applied to spatial GEV models, consists of two main steps:

1. **Max Step**: Maximum likelihood estimation of GEV parameters at each spatial location
2. **Smooth Step**: Spatial smoothing of the maximum likelihood estimates using a latent Gaussian field

The algorithm treats the ML estimates as sufficient statistics for a latent Gaussian field, providing a fast approximation to full Bayesian inference for spatial extreme value models.

Code Structure

The package is organized into several key files:

1. `src/max.cpp`: Implements the Max step (maximum likelihood estimation for GEV)
2. `src/smooth.cpp`: Implements the Smooth step (latent Gaussian field smoothing)
3. `src/maxandsmooth.cpp`: Ties together the Max and Smooth steps
4. `src/gev.cpp`: Implements GEV-specific functions for likelihood and gradient calculations

Methods

Max Step

The Max step involves computing location-wise maximum likelihood estimates (MLEs) for the GEV model parameters. This step is performed independently for each location, treating the data as if it were independent across locations.

Data Structure

The input data Y, calculated from the UKCP, is structured as a matrix, where:

- Rows represent observations of hourly maximum rainfall in yearly blocks
- Columns represent different spatial locations over Great Britain.

Maximum Likelihood Estimation

For each location (column in Y), we compute the MLEs for the three GEV parameters: location (μ), scale (σ), and shape (ξ). The log-likelihood function for the GEV distribution at a single location is:

$$\ell(\mu, \sigma, \xi | y) = -n \log \sigma - (1 + \frac{1}{\xi}) \sum_{i=1}^{n} \log \left(1 + \xi \frac{y_i - \mu}{\sigma}\right) - \sum_{i=1}^{n} \left(1 + \xi \frac{y_i - \mu}{\sigma}\right)^{-1/\xi}$$

where n is the number of observations at the location.

The MLEs are obtained by maximizing this likelihood function with respect to the parameters:

$$(\hat{\mu}_i, \hat{\sigma}_i, \hat{\xi}_i) = \arg\max_{(\mu,\sigma,\xi)} \ell(\mu, \sigma, \xi | Y_i)$$

where Y_i is the data for location i.

Link Functions

Instead of directly maximizing the likelihood with the original parameters, we transform the parameters using a link function

$$(\psi, \tau, \phi) = h(\mu, \sigma, \xi) = (\log(\mu), \log(\sigma) - \log(\mu), \text{logit}(\xi))$$

Implementation

The maximization is performed using numerical optimization techniques. In our C++ implementation, we use the following approach:

1. Parameters are transformed with a link function
2. The negative log-likelihood and its gradient are computed on the unconstrained scale
3. A numerical optimizer is used to find the MLEs as well as the Hessians at each location's optimum.

The output of this step includes:

1. A vector of parameter estimates, $\hat{\eta}$, ordered such that each station's location parameter appears first, followed by scale parameters, then shape parameters:

   $$\hat{\eta} = (\hat{\psi}_1, \dots, \hat{\psi}_n, \hat{\tau}_1, \dots, \hat{\tau}_n, \hat{\phi}_1, \dots, \hat{\phi}_n)^T$$

   where n is the number of stations.

2. A precision matrix, $Q_{\eta y}$, constructed from the negative Hessians of the log-likelihood at the MLE estimates. Due to the parameter ordering in $\hat{\eta}$, $Q_{\eta y}$ can be described as a 3×3 block matrix:

   $$Q_{\eta y} = \begin{bmatrix} Q_{\psi\psi} & Q_{\psi\tau} & Q_{\psi\phi} \\ Q_{\tau\psi} & Q_{\tau\tau} & Q_{\tau\phi} \\ Q_{\phi\psi} & Q_{\psi\tau} & Q_{\phi\phi} \end{bmatrix}$$

where each block Q_{ij} is an $n \times n$ diagonal matrix. The diagonal elements of Q_{ii} correspond to the negative second derivatives of the log-likelihood with respect to the i-th parameter at each station. The off-diagonal blocks Q_{ij} (where $i \neq j$) contain the negative mixed partial derivatives of the log-likelihood with respect to the i-th and j-th parameters.

For example, the elements of $Q_{\psi\psi}$ are the conditional precisions of the location parameters:

$$
\begin{aligned}
Q_{\psi\psi} &= \operatorname{diag}\left(\tau_1^{\psi\psi}, \dots, \tau_n^{\psi\psi}\right) \\
&= \operatorname{diag}\left(-\frac{\partial^2 \ell(Y_1|\psi_1, \tau_1, \phi_1)}{\partial \psi_1^2}, \dots, -\frac{\partial^2 \ell(Y_n|\psi_n, \tau_n, \phi_n)}{\partial \psi_n^2}\right),
\end{aligned}
$$

and the elements of $Q_{\mu\xi}$ are the conditional dependencies between the location and shape parameters:

$$
\begin{aligned}
Q_{\psi\phi} &= \operatorname{diag}\left(\tau_1^{\psi\phi}, \dots, \tau_n^{\psi\phi}\right) \\
&= \operatorname{diag}\left(-\frac{\partial^2 \ell(Y_1|\psi_1, \tau_1, \phi_1)}{\partial \psi_1 \partial \phi_1}, \dots, -\frac{\partial^2 \ell(Y_n|\psi_n, \tau_n, \phi_n)}{\partial \psi_n \partial \phi_n}\right)
\end{aligned}
$$

This structure reflects the independence assumption between stations in the Max step, while capturing the parameter dependencies within each station. The outputs, $\hat{\eta}$ and $Q_{\eta y}$, serve as inputs into the Smooth step.

Smooth Step

The Smooth step involves Bayesian inference on a latent Gaussian field, using the maximum likelihood estimates from the Max step as data.

Model Structure

Let η be the latent field of parameters, and $\hat{\eta}$ be the maximum likelihood estimates from the Max step. The model is structured as follows:

1. Data level: $\hat{\eta}|\eta \sim N(\eta, Q_{\eta y}^{-1})$
2. Latent level: $\eta|\tau_\eta \sim N(0, Q_\eta^{-1})$
3. Hyperparameter level: $\tau_\eta \sim \pi(\tau_\eta)$

where $Q_{\eta y}$ is the precision matrix from the Max step, and Q_η is the precision matrix of the latent field, parameterized by hyperparameters τ_η.

Data Level

The data level models the relationship between the observed maximum likelihood estimates $\hat{\eta}$ and the true latent field η:

$$\hat{\eta}|\eta \sim N(\eta, Q_{\eta y}^{-1})$$

Here, $Q_{\eta y}$ is the precision matrix obtained from the Max step, representing the uncertainty in the MLE estimates. This can be thought of as using the outputs of the Max step as sufficient statistics for the observed data.

Latent Level

The latent level models the spatial structure of the parameter field:

$$\eta|\tau_\eta \sim N(0, Q_\eta^{-1})$$

Here, Q_η is a block diagonal precision matrix that encodes the spatial dependence structure of the latent field. The block diagonal structure reflects the assumption of independence between different parameters, while allowing for spatial correlation within each parameter field. The precision matrix Q_η is constructed as follows:

$$Q_\eta = \begin{bmatrix} \tau_\psi Q_{\text{prior}} & 0 & 0 \\ 0 & \tau_\tau Q_{\text{prior}} & 0 \\ 0 & 0 & \tau_\phi Q_{\text{prior}} \end{bmatrix}$$

where

$$Q_{\text{prior}} = \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \vdots & \vdots & \vdots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix}$$

and $\tau_\eta = \left(\tau_\psi, \tau_\tau, \tau_\phi\right)^T$ are the hyperparameters controlling the strength of spatial dependence for each parameter.

Hyperparameter Level

The hyperparameters τ_η are assigned prior distributions.

Posterior Distribution

The posterior distribution of interest is:

$$p(\eta, \tau | \hat{\eta}) \propto p(\hat{\eta} | \eta) p(\eta | \tau_\eta) p(\tau_\eta)$$

MCMC Sampling

The Smooth step uses Metropolis-Hastings MCMC to sample from this posterior. The algorithm alternates between:

1. Proposing new hyperparameters τ_η^*
2. Sampling a new latent field η^* conditional on τ_η^*
3. Accepting or rejecting the proposal based on the Metropolis-Hastings ratio

Latent Field Sampling

The latent field is sampled from the conditional normal distribution

$$\eta | \hat{\eta}, \tau_\eta \sim \mathrm{N}(\mu_{\mathrm{post}}, Q_{\mathrm{post}}),$$

where

$$Q_{\mathrm{post}} = Q_{\eta y} + Q_\eta \qquad \text{and} \qquad \mu_{\mathrm{post}} = Q_{\mathrm{post}}^{-1} Q_{\eta y} \hat{\eta}.$$

The first thing we can do to make these calculations faster is to precompute $\hat{b}_{\eta y} = Q_{\eta y} \hat{\eta}$ between the Max step and the Smooth step, giving us

$$Q_{\mathrm{post}} = Q_{\eta y} + Q_\eta \qquad \text{and} \qquad \mu_{\mathrm{post}} = Q_{\mathrm{post}}^{-1} \hat{b}_{\eta y}.$$

The most efficient way to sample from this posterior is to calculate the Cholesky decomposition

$$Q_{\mathrm{post}} = L_{\mathrm{post}} L_{\mathrm{post}}^T,$$

and using that to solve for μ_{post} in

$$L_{\mathrm{post}} L_{\mathrm{post}}^T \mu_{\mathrm{post}} = \hat{b}_{\eta y}.$$

Cholesky Decomposition of Q_{post}

We first write out Q_{post} as

$$Q_{\text{post}} = \begin{bmatrix} Q_{\psi\psi} + \tau_\psi Q_{\text{prior}} & Q_{\psi\tau} & Q_{\psi\phi} \\ Q_{\tau\psi} & Q_{\tau\tau} + \tau_\tau Q_{\text{prior}} & Q_{\tau\phi} \\ Q_{\phi\psi} & Q_{\phi\tau} & Q_{\phi\phi} + \tau_\phi Q_{\text{prior}}, \end{bmatrix}$$

where each block on the diagonal has bandwidth 3 and all the off-diagonal blocks are diagonal matrices. Keep in mind that since the off-diagonal blocks are diagonal matrices we have that $Q_{\psi\tau} = Q_{\tau\psi}$, $Q_{\psi\phi} = Q_{\phi\psi}$ and $Q_{\tau\phi} = Q_{\phi\tau}$.We can thus calculate the Cholesky decomposition

$$L_{\text{post}} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33}, \end{bmatrix}$$

efficiently using a block-Cholesky factorization:

1. $L_{11} = \text{Chol}(Q_{\psi\psi} + \tau_\psi Q_{\text{prior}})$
2. $L_{21} = Q_{\psi\tau} L_{11}^{-T}$
3. $L_{31} = Q_{\psi\phi} L_{11}^{-T}$
4. $L_{22} = \text{Chol}(S_{22})$
   - $S_{22} = \left(Q_{\tau\tau} + \tau_\tau Q_{\text{prior}}\right) - L_{21} L_{21}^T$
5. $L_{32} = \left(Q_{\phi\tau} - L_{31} L_{21}^T\right) L_{22}^{-T}$
6. $L_{33} = \text{Chol}(S_{33})$
   - $S_{33} = \left(Q_{\phi\phi} + \tau_\phi Q_{\text{prior}}\right) - L_{31} L_{31}^T - L_{32} L_{32}^T$