

# Gaussian Copulas for Large Spatial Fields

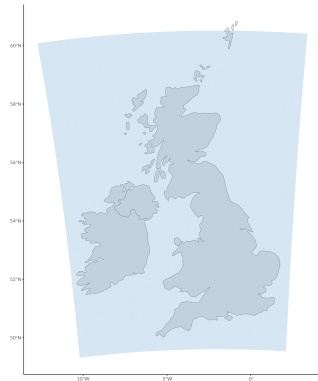
Modeling Data-Level Spatial Dependence in Multivariate Generalized Extreme Value Distributions

Brynjólfur Gauti Guðrúnar Jónsson

University of Iceland

# Introduction

- ▶ UKCP Local Projections on a 5km grid over the UK (1980-2080)
- ▶ Challenge: Modeling maximum daily precipitation in yearly blocks
  - ▶ 43,920 spatial locations on a  $180 \times 244$  grid
- ▶ Two aspects of spatial dependence:
  1. GEV parameters (ICAR models)
  2. Data-level dependence (Copulas)



# Calculating Multivariate Normal Densities

## Log Density Formula

$$\log f(\mathbf{x}) \propto \frac{1}{2} (\log |\mathbf{Q}| - \mathbf{x}^T \mathbf{Q} \mathbf{x})$$

## Key Components

1. **Log Determinant:**  $\log |\mathbf{Q}|$ 
  - ▶ Constant for a given precision matrix
2. **Quadratic Form:**  $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ 
  - ▶ Needs calculation for each density evaluation

## Computational Challenges

- ▶ Log determinant calculation
  - ▶ Time complexity:  $O(n^3)$  for naive methods
  - ▶ Memory complexity:  $O(n^2)$
- ▶ Quadratic form calculation
  - ▶ Time complexity:  $O(n^2)$
  - ▶ Critical for performance in large spatial fields

## Spatial Model Considerations

- ▶ Some models (e.g., ICAR) avoid log determinant calculation
- ▶ Efficient computation crucial for large-scale applications

# Spatial Models

## Conditional Autoregression (CAR)

- ▶  $\mathbf{D}$  is a diagonal matrix with  $D_{ii} = n_i$ , the number of neighbours of  $i$
- ▶  $\mathbf{A}$  is the adjacency matrix with  $A_{ij} = A_{ji} = 1$  if  $i \sim j$
- ▶  $\tau$  models overall precision

$$\mathbf{x} \sim N(\mathbf{0}, \tau \mathbf{Q})$$

$$\mathbf{Q} = \mathbf{D} (\mathbf{I} - \alpha \mathbf{A})$$

## Besag's Intrinsic Conditional Autoregression (ICAR)

- ▶  $\alpha = 1$ , so  $\mathbf{Q}$  is singular, but constant
- ▶ Don't have to calculate  $\log |\mathbf{Q}|$
- ▶  $\tau$  is a precision parameter

$$\mathbf{x} \sim N(\mathbf{0}, \tau \mathbf{Q})$$

$$\mathbf{Q} = \mathbf{D} - \mathbf{W}$$

# Spatial Models

## BYM (Besag-York-Mollié) Model

- ▶  $\mathbf{u}$  is the structured spatial component (Besag model)
- ▶  $\mathbf{v}$  is the unstructured component (i.i.d. normal)
- ▶  $\tau_u$  and  $\tau_v$  are precision parameters for each component

$$\mathbf{x} = \mathbf{u} + \mathbf{v}$$

$$\mathbf{u} \sim \text{ICAR}(\tau_u)$$

$$\mathbf{v} \sim N(\mathbf{0}, \tau_v^{-1})$$

## BYM2 Model

- ▶ Rewrite the combination to get proper scaling
- ▶  $\rho$  models how much of variance is spatial
- ▶  $s$  is a scaling factor chosen to make  $\text{Var}(\mathbf{u}_i) \approx 1$

$$\mathbf{x} = \left( \left( \sqrt{\rho/s} \right) \mathbf{u} + \left( \sqrt{1-\rho} \right) \mathbf{v} \right) \sigma$$

$$\mathbf{u} \sim \text{ICAR}(1)$$

$$\mathbf{v} \sim N(\mathbf{0}, n)$$

# Spatial Modeling on Parameter-level

- ▶  $\mu = \mu_0 (1 + \Delta (t - t_0))$ , location
- ▶  $\sigma$ : scale
- ▶  $\xi$ : shape

$$\log(\mu_0) = \psi \sim \text{BYM2}(\mu_\psi, \rho_\psi, \sigma_\psi)$$

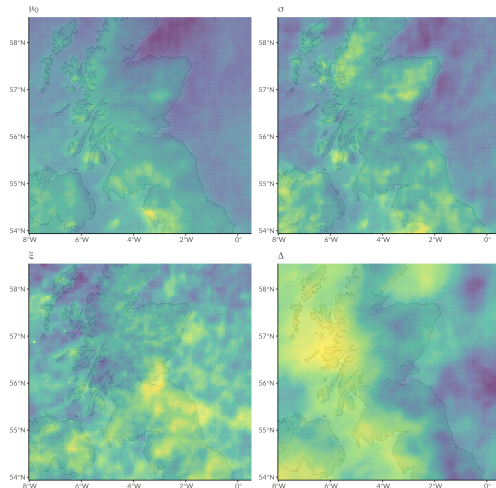
$$\log(\mu_0) - \log(\sigma) = \tau \sim \text{BYM2}(\mu_\tau, \rho_\tau, \sigma_\tau)$$

$$f_\xi(\xi) = \phi \sim \text{BYM2}(\mu_\phi, \rho_\phi, \sigma_\phi)$$

$$f_\Delta(\Delta) = \gamma \sim \text{BYM2}(\mu_\gamma, \rho_\gamma, \sigma_\gamma)$$

Our BYM2 hyperparameters point to a large degree of spatial variation									
variable	mean	median	sd	mad	q5	q95	rhat	ess_bulk	ess_tail
$\sigma_\psi$	0.072	0.072	0.001	0.001	0.070	0.074	1.012	137	387
$\mu_\psi$	2.133	2.133	0.001	0.001	2.131	2.135	1.001	5,029	3,417
$\rho_\psi$	0.998	0.998	0.001	0.001	0.997	0.999	1.000	3,161	3,218
$\sigma_\tau$	0.102	0.102	0.002	0.002	0.098	0.106	1.015	381	973
$\mu_\tau$	-0.923	-0.923	0.001	0.001	-0.926	-0.921	1.000	4,130	3,507
$\rho_\tau$	0.997	0.998	0.001	0.001	0.996	0.999	1.001	3,192	2,903
$\sigma_\phi$	0.358	0.358	0.009	0.009	0.343	0.372	1.008	401	730
$\mu_\phi$	0.341	0.341	0.004	0.004	0.335	0.347	1.001	3,728	3,049
$\rho_\phi$	0.996	0.997	0.001	0.001	0.994	0.998	1.002	1,995	2,677
$\sigma_\gamma$	0.332	0.333	0.011	0.011	0.313	0.351	1.029	106	208
$\mu_\gamma$	1.438	1.438	0.012	0.012	1.419	1.458	1.001	2,517	3,053
$\rho_\gamma$	0.996	0.996	0.002	0.001	0.993	0.998	1.000	3,823	2,875

Spatial distribution of posterior means  
GEV parameters on constrained scales



# From Parameter-level to Data-level Dependence

## Parameter-level Dependence

- ▶ Assumes conditional independence
- ▶ Biased joint probability estimates
- ▶ Underestimates parameter variance

## Copula

- ▶ Improves joint probabilities
- ▶ Enhances spatial risk assessment
- ▶ Better variance estimates

**Sklar's Theorem:** For any multivariate distribution  $H$ , there exists a unique copula  $C$  such that:

$$H(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d))$$

where  $F_i$  are marginal distributions. We can also write this as a log-density

$$\log h(x) = \log c(F_1(x_1), \dots, F_d(x_d)) \sum_{i=1}^d \log f_i(x_i)$$

## Our Approach: Matérn-like Gaussian Copula

Marginal CDFs,  $F_i(x_i)$ , is GEV( $\mu_i, \sigma_i, \xi_i$ )

$$\log h(\mathbf{x}) = \log c(u_1, \dots, u_d) + \sum_{i=1}^d f_{\text{GEV}}(x_i | \mu_i, \sigma_i, \xi_i)$$

$$u_i = F_{\text{GEV}}(x_i | \mu_i, \sigma_i, \xi_i)$$

### Gaussian Copula

$$\begin{aligned} \log c(\mathbf{u}) &\propto \frac{1}{2} (\log |\mathbf{Q}| - \mathbf{z}^T \mathbf{Q} \mathbf{z} + \mathbf{z}^T \mathbf{z}) \\ \mathbf{z} &= \Phi^{-1}(\mathbf{u}) \end{aligned}$$



## The Precision Matrix

$\mathbf{Q}$  defined as Kronecker sum of two AR(1) precision matrices

$$\mathbf{Q} = \left( \mathbf{Q}_{\rho_1} \otimes \mathbf{I}_{n_2} + \mathbf{I}_{n_1} \otimes \mathbf{Q}_{\rho_2} \right)^{\nu+1}, \quad \nu \in \{0, 1, 2\}$$

$$\mathbf{Q}_{\rho} = \frac{1}{1 - \rho^2} \begin{bmatrix} 1 & -\rho & 0 & \cdots & 0 \\ -\rho & 1 + \rho^2 & -\rho & \cdots & 0 \\ 0 & -\rho & 1 + \rho^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

## Eigendecomposition

Because of how  $\mathbf{Q}$  is defined, we know that

$$\begin{aligned}\mathbf{Q} &= \mathbf{V}\mathbf{\Lambda}\mathbf{V} \\ &= (\mathbf{V}_1 \otimes \mathbf{V}_2)(\mathbf{\Lambda}_{\rho_1} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{\Lambda}_{\rho_2})^{\nu+1}(\mathbf{V}_1 \otimes \mathbf{V}_2)^T \\ \mathbf{Q}_{\rho_1} &= \mathbf{V}_1 \mathbf{\Lambda}_{\rho_1} \mathbf{V}_1^T \quad \& \quad \mathbf{Q}_{\rho_2} = \mathbf{V}_2 \mathbf{\Lambda}_{\rho_2} \mathbf{V}_2^T\end{aligned}$$

Spectral decomposition defined by value/vector pairs of smaller matrices

$$\left\{ \lambda_{\rho_1} \right\}_i + \left\{ \lambda_{\rho_2} \right\}_j \qquad \left\{ \mathbf{v}_{\rho_1} \right\}_i \otimes \left\{ \mathbf{v}_{\rho_2} \right\}_j$$

- ▶ Problem:  $\Sigma_{ii} = (\mathbf{Q}^{-1})_{ii} \neq 1$
- ▶ Solution:  $\tilde{\mathbf{Q}} = \mathbf{D}\mathbf{Q}\mathbf{D}$ , where  $\mathbf{D}_{ii} = \sqrt{\Sigma_{ii}}$

## Marginal Standard Deviations

$$\Sigma = \mathbf{Q}^{-1} = (\mathbf{V}\Lambda\mathbf{V}^T)^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{V}$$

We know that if  $A = BC$  then  $A_{ii} = B_{i,\cdot}C_{\cdot,i}$ , so

$$\Sigma_{ii} = \sum_{k=1}^n v_{ik} \frac{1}{\lambda_k} (v^T)_{ki} = \sum_{k=1}^n v_{ik} \frac{1}{\lambda_k} v_{ik} = \sum_{k=1}^n v_{ik}^2 \frac{1}{\lambda_k}$$

Compute vector  $\sigma^2$  containing all marginal variances

$$\sigma^2 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \frac{\left( \left\{ \mathbf{v}_{\rho_1} \right\}_i \otimes \left\{ \mathbf{v}_{\rho_2} \right\}_j \right)^2}{\left( \left\{ \lambda_{\rho_1} \right\}_i + \left\{ \lambda_{\rho_2} \right\}_j \right)^{\nu+1}}$$

# Marginal Standard Deviations

```
dim1 <- 50; dim2 <- 50
rho1 <- 0.5; rho2 <- 0.3
nu <- 2
Q1 <- make_AR_prec_matrix(dim1, rho1)
Q2 <- make_AR_prec_matrix(dim2, rho2)
I1 <- Matrix::Diagonal(dim1)
I2 <- Matrix::Diagonal(dim2)
Q <- temp <- kronecker(Q1, I2) + kronecker(I1, Q2)
for (i in seq_len(nu)) Q <- Q %**% temp
```

```
msd <- function(Q1, Q2) {
  E1 <- eigen(Q1)
  E2 <- eigen(Q2)
  marginal_sd_eigen(
    E1$values, E1$vectors, dim1,
    E2$values, E2$vectors, dim2,
    nu
  ) |> sort()
}
```

```
bench::mark(
  "solve" = solve(Q) |> diag() |> sqrt() |> sort(),
  "inla.qinv" = inla.qinv(Q) |> diag() |> sqrt() |> sort(),
  "marginal_sd_eigen" = msd(Q1, Q2),
  iterations = 10, filter_gc = FALSE
)
```

# A tibble: 3 x 6

	expression	min	median	`itr/sec`	mem_alloc	`gc/sec`
	<bch:expr>	<bch:tm>	<bch:tm>	<dbl>	<bch:byt>	<dbl>
1	solve	1.27s	1.29s	0.757	78.17MB	0.757
2	inla.qinv	358.2ms	367.86ms	2.72	4.33MB	0
3	marginal_sd_eigen	3.29ms	3.35ms	279.	649.35KB	0