Bridging Gaussian Markov Random Fields and Copulas: A Fast Algorithm for Efficient Gaussian Copula Density Computation with Matérn-like Precision Matrices

Brynjólfur Gauti Guðrúnar Jónsson*

University of Iceland,

E-mail: bryjolfur@hi.is

Abstract

Gaussian Markov Random Fields (GMRFs) have long been a powerful tool for modeling spatial and temporal dependencies in various fields. Similarly, copulas have proven invaluable for modeling complex dependency structures in multivariate data. However, the combination of these two approaches - using GMRFs within copula models - has historically been computationally inefficient, limiting their joint application to smaller datasets or simpler models. This work presents a novel algorithm that overcomes these limitations, allowing for fast and efficient computation of Gaussian Copula densities using GMRF precision structures. By bridging the gap between GMRFs and copulas, this method opens up new possibilities for analyzing large-scale, high-dimensional spatial and spatio-temporal data with complex dependency structures.

Introduction

Review

Gaussian Markov Random Fields (GMRFs) and copulas are two powerful statistical tools, each offering unique strengths in modeling complex data structures. GMRFs excel in capturing spatial and temporal dependencies, particularly in fields such as environmental science, epidemiology, and image analysis. Their ability to represent local dependencies through sparse precision matrices makes them computationally attractive for high-dimensional problems. Copulas, on the other hand, provide a flexible framework for modeling multivariate dependencies, allowing separate specification of marginal distributions and their joint behavior.

The Gaussian copula, in particular, has gained popularity due to its interpretability and connection to the multivariate normal distribution. However, combining GMRFs with copulas has historically been computationally challenging, limiting their joint application to smaller datasets or simpler models.

Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a multivariate random vector with marginal distribution functions F_i for $i = 1, 2, \ldots, n$. The joint distribution function of $\mathbf{X}$ can be written as:

$$F_{\mathbf{X}}(\mathbf{x}) = C(F_1(x_1), F_2(x_2), \ldots, F_n(x_n)),$$

where C is the Gaussian copula defined by the GMRF precision matrix $\mathbf{Q}$. The Gaussian copula C is given by:

$$C(u_1, u_2, \ldots, u_n) = \Phi_{\mathbf{Q}}(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \ldots, \Phi^{-1}(u_n)),$$

where $\Phi_{\mathbf{Q}}$ is the joint cumulative distribution function of a multivariate normal distribution with mean vector $\mathbf{0}$ and precision matrix $\mathbf{Q}$, and Φ^{-1} is the inverse of the standard normal cumulative distribution function.

A critical requirement for the precision matrix $\mathbf{Q}$ governing the GMRF copula C is that $= \mathbf{Q}^{-1}$ should have a unit diagonal, i.e. the marginal variance is equal to one everywhere.. This ensures it operates on the same scale as the transformed data, $\Phi^{-1}(u_i)$. However, this can be challenging as GMRFs are typically defined in terms of precision matrices that often imply non-unit marginal variances.

This paper presents a novel algorithm that bridges the gap between GMRFs and copulas, allowing for fast and efficient computation of Gaussian copula densities using GMRF precision structures. Our method focuses on creating a Matérn-like precision matrix $\mathbf{Q}$ with unit marginal variance and efficiently computing the multivariate Gaussian copula density of $\mathbf{Z} = \Phi^{-1}(\mathbf{u})$, where $u_i \sim \text{Uniform}(0, 1)$, $i = 1, \dots, n$.

The key innovation lies in leveraging the special structure of the precision matrix:

$$\mathbf{Q} = \mathbf{Q}_1 \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{Q}_1,$$

where $\mathbf{Q}_1$ is the precision matrix of a standardized one-dimensional AR(1) process and $\otimes$ denotes the Kronecker product. By employing efficient eigendecomposition techniques, our method avoids explicit formation and inversion of the large precision matrix $\mathbf{Q}$, making it particularly suitable for high-dimensional spatial data. In additions to the exact method, we mention approximations to $\mathbf{Q}$ using circuland and folded circulant matrices.

Problem Formulation

Consider a spatial field on a regular $n \times n$ grid. Our objective is to compute the Gaussian copula density efficiently for this field. This computation involves:

1. Specifying a precision matrix $\mathbf{Q}$ that represents the spatial dependence structure.

2. Ensuring the implied covariance matrix $\quad = \mathbf{Q}^{-1}$ has unit diagonal elements.
3. Computing the density for large spatial fields in a computationally efficient manner.

Precision Matrix Structure

We use a Matérn-like precision matrix structure defined as:

$$
\begin{aligned}
\mathbf{Q} &= \tilde{\mathbf{Q}}\ , \quad \nu \in \{0, 1, 2\} \\
\tilde{\mathbf{Q}} &= \mathbf{Q}_1 \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{Q}_1,
\end{aligned}
$$

where $\mathbf{Q}_1$ is the precision matrix of a standardized one-dimensional AR(1) process, and $\otimes$ denotes the Kronecker product.

Eigendecomposition Method

The properties of Kronecker products and sums are utilized to compute the eigendecomposition of $\mathbf{Q}$. This approach facilitates:

1. Calculation of $\log |\mathbf{Q}|$.
2. Computation of quadratic forms $\mathbf{x}^T \mathbf{Q} \mathbf{x}$.
3. Determination of the diagonal elements of $\quad = \mathbf{Q}^{-1}$ without explicit matrix inversion.

Unit Marginal Variance Adjustment

To ensure unit marginal variances, which is necessary for copula applications, we employ a scaling method for the precision matrix.

Approximation Methods

We introduce circulant and folded circulant approximations to $\mathbf{Q}$. These approximations offer potential computational advantages, which we analyze in terms of efficiency and accuracy trade-offs.

Methods

Gaussian Copula Density Computation

The Gaussian copula density for a random vector $\mathbf{U} = (U_1, ..., U_n)$ with $U_i \sim \text{Uniform}(0, 1)$ is given by:

$$c(\mathbf{u}) = |\mathbf{Q}|^{1/2} \exp\left(-\frac{1}{2}\mathbf{z}^T(\mathbf{Q} - \mathbf{I})\mathbf{z}\right)$$

where $\mathbf{z} = (z_1, ..., z_n)$ with $z_i = \Phi^{-1}(u_i)$, $\mathbf{Q}$ is the precision matrix, and $\mathbf{I}$ is the identity matrix.

The log-density can be expressed as:

$$\log c(\mathbf{u}) = \frac{1}{2}\log|\mathbf{Q}| - \frac{1}{2}\mathbf{z}^T\mathbf{Q}\mathbf{z} + \frac{1}{2}\mathbf{z}^T\mathbf{z}$$

Our goal is to efficiently compute this log-density for large spatial fields.

Precision Matrix Structure

We define the precision matrix $\mathbf{Q}$ as:

$$\mathbf{Q} = (\mathbf{Q}_1 \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{Q}_1)^{\nu}, \quad \nu \in \{0, 1, 2\}$$

where $\mathbf{Q}_1$ is the precision matrix of a one-dimensional AR(1) process. This matrix is then scaled so that it implied unit marginal variances.

Computation Process

Step 1: Eigendecomposition of $\mathbf{Q}_1$

We first compute the eigendecomposition of $\mathbf{Q}_1$:

$$\mathbf{Q}_1 = \mathbf{V} \ \mathbf{V}^T$$

where $\mathbf{V}$ is the matrix of eigenvectors and   is the diagonal matrix of eigenvalues. Then, the eigendecomposition of $\mathbf{Q}$ is:

$$\mathbf{Q} = (\mathbf{V} \otimes \mathbf{V})(\ \otimes \mathbf{I} + \mathbf{I} \otimes \ )^{\nu} (\mathbf{V} \otimes \mathbf{V})^T.$$

This means that the eigenvectors of $\mathbf{Q}$ are $\mathbf{v}_j \otimes \mathbf{v}_i$ and the corresponding eigenvalues are $\lambda_i + \lambda_j$.

Step 2: Computation of Marginal Standard Deviations

Using the eigendecomposition of $\mathbf{Q}_1$, we compute the marginal standard deviations and store them in a vector , i.e. $\sigma_i = \sqrt{\Sigma_{ii}}$:

$$= \sqrt{\sum_{i,j} \frac{(\mathbf{v}_j \otimes \mathbf{v}_i)^2}{(\lambda_i + \lambda_j)^{\nu}}}$$

where $\mathbf{v}_i$ are the eigenvectors and λ_i are the eigenvalues of $\mathbf{Q}_1$.

Step 3: Scaling the Eigendecomposition

We scale the eigendecomposition of $\mathbf{Q}$ using the marginal standard deviations:

$$
\begin{aligned}
\mathbf{DQD} &= \mathbf{D}(\mathbf{V} \otimes \mathbf{V})(\quad \otimes \mathbf{I} + \mathbf{I} \otimes \quad)^{\nu}(\mathbf{V} \otimes \mathbf{V})^{T}\mathbf{D} \\
&= (\tilde{\mathbf{V}} \otimes \tilde{\mathbf{V}})(\tilde{}\quad \otimes \mathbf{I} + \mathbf{I} \otimes \tilde{}\quad)^{\nu}(\tilde{\mathbf{V}} \otimes \tilde{\mathbf{V}})^{T}
\end{aligned}
$$

where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sigma_i$.

Step 4: Efficient Computation of Log-Density

Using this scaled eigendecomposition, we efficiently compute:

1. Log-determinant: $\log |\mathbf{DQD}| = \sum_{i,j} \nu \log(\tilde{\lambda}_i + \tilde{\lambda}_j)$
2. Quadratic form: $\mathbf{z}^T\mathbf{DQDz} = \sum_{i,j} (\tilde{\lambda}_i + \tilde{\lambda}_j)^{\nu} y_{ij}^2$, where $y_{ij} = (\tilde{\mathbf{v}}_j \otimes \tilde{\mathbf{v}}_i)^T\mathbf{z}$

This approach allows for efficient computation of the Gaussian copula density without explicitly forming the full $n^2 \times n^2$ precision matrix $\mathbf{Q}$ or its inverse .