ARTICLE TEMPLATE

A Vecchia approximation for high-dimensional Gaussian cumulative distribution functions arising from spatial data

ARTICLE HISTORY
Compiled August 2, 2022

Abstract
We introduce an approach to quickly and accurately approximate the cumulative distribution function of multivariate Gaussian distributions arising from spatial Gaussian processes. This approximation is trivially parallelizeable and simple to implement using standard software. We demonstrate its accuracy and computational efficiency in a series of simulation experiments, and apply it to analyzing the joint tail of a large precipitation dataset using a recently-proposed scale mixture model for spatial extremes. This dataset is many times larger than what was previously considered possible to fit using preferred inferential techniques.

KEYWORDS
Gaussian process ; Scale mixture ; Spatial extremes

1. Introduction

We introduce a trivially parallelizable approach to quickly and accurately approximate the cumulative distribution function (*cdf*) of multivariate Gaussian distributions with highly structured covariance matrices, such as those arising from spatial Gaussian processes. The multivariate Gaussian distribution is by far the most widely used for modeling multivariate and spatial data. To a large degree, its near universal adoption is the result of its simplicity; it is concisely and intuitively parametrized by a mean vector and pairwise dependence in the form of a covariance matrix. Prominent examples of its use include time series models like autoregressive and moving average models, which consider the joint distribution of the observations taken at discrete time points to be multivariate Gaussian, as well as geostatistics models, which consider spatially-indexed observations to be realizations of a Gaussian process, usually with a parsimoniously parametrized covariance structure. Even multivariate models that do not assume Gaussian responses often represent dependence using some kind of latent multivariate Gaussian distribution.

In most situations, likelihood-based inference on popular models just requires calculation of the joint density of all observations. The probability density function (*pdf*) for a multivariate Gaussian random variable is

$$f(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \equiv \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) = (2\pi)^{-k/2}\det(\boldsymbol{\Sigma})^{-1/2}\exp\Big[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\Big], \quad (1)$$

where $\boldsymbol{\mu}$ is the mean vector of length D and $\boldsymbol{\Sigma}$ is the $D \times D$ covariance matrix.

In principle, there is nothing difficult about calculating this density; it simply requires commonplace operations like calculating an exponent, matrix determinant, ma-

trix multiplication, and matrix inversion. However this is not an easy task in practice when the dimension D is large. The complexity of calculating the determinant and inverse of a $D \times D$ matrix is typically $O(D^3)$ for algorithms in common use. This means that for large values of D, the calculation of the *pdf* becomes prohibitive.

Computing the Gaussian *cdf*, which is a much more difficult problem, has received much less attention. The problem has increased in prominence recently with advances in spatial modeling of extreme events. State-of-the-art approaches for spatial extremes like Wadsworth and Tawn [1], Thibaud et al. [2], de Fondeville and Davison [3], and Huser and Wadsworth [4] all require high-dimensional Gaussian *cdf*s for inference. This turns out to be the dominant computational bottleneck, and all but de Fondeville and Davison [3] restricted their analyses to fewer than 20 spatial locations because larger datasets are computationally intractable using widely-used techniques for computing the Gaussian *cdf*. In real-world spatial applications, one should expect to see many more spatial locations, and existing approaches are not equipped to handle datasets of even moderate size.

Multivariate Gaussian *cdf*s appear in other contexts as well; for example the density of multivariate skewed Gaussian and t random variables are functions of the multivariate Gaussian *cdf* [5]. Here, we will focus on the case of spatial extremes. To make things concrete, we will use the example of the Gaussian scale mixture model from [6], although our computational strategy would work equally well in any context with highly-structured covariance matrices.

Most approaches to calculating multivariate Gaussian probabilities are intended for problems of small or moderate dimension. Genz [7] proposed a transformation from the original integral over $\mathbb{R}^D$ to an integral over a unit hypercube. Transforming to a finite region then allows the use any standard numerical integration method. Genz [8] derived formulas to calculate bivariate and trivariate Gaussian *cdf*s with high precision using Gauss-Legendre numerical integration. The calculations are fast and precise but do not apply in higher dimensions. Miwa et al. [9] proposed a two-stage recursive approach to estimate the Gaussian *cdf*. Their approach does not scale to high dimensions because it requires a sum over a combinatorially exploding (in D) number of terms.

The most popular approach for approximating Gaussian *cdf*s in moderate dimensions was proposed by Genz and Bretz [10]. They describe the use of Monte Carlo (MC) and quasi-Monte Carlo (QM) methods to estimate the joint *cdf*. Their QM methods have smaller asymptotic errors than the MC versions, and hence are the more widely used.

More recently, Genton et al. [11] sped up the Genz and Bretz [10] QM algorithm by performing matrix computations with fast hierarchical matrix libraries [12]. As a follow-up, Cao et al. [13] combined hierarchical matrix computations with a blocking technique to further speed up computations. These approaches are much faster than their predecessors and work for Gaussian random variables with arbitrary covariance structures. They lean heavily on linking to specialized libraries for matrix operations. Our approach achieves speedups using a fundamentally different strategy, by specifically leveraging the properties of highly-structured covariance forms arising from, for example, time series or spatial data. It requires no exotic software, and is trivially parallelizable using simple tools in `R`.

2. A Vecchia Approximation for the Multivariate Gaussian Distribution Function

The multivariate Gaussian *cdf* that we wish to calculate is simply the integral of the *pdf* (1),

$$P(\mathbf{X} < \mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv \Phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_k} \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{y})\, dy_1 \ldots dy_D. \tag{2}$$

To calculate the integral (2), one must resort to numerical techniques, as it is well-known that no closed form exists, even in a single dimension. In high dimensions, numerical integration is very difficult simply due to geometry and the curse of dimensionality. The difficulty is compounded in the case of the Gaussian *cdf* because while the curse of dimensionality requires an exponentially (in D) increasing number of evaluations of the integrand, the cost of each evaluation of the integration itself grows as D^3. We seek a technique that simultaneously 1) reduces the effective dimension of the integral and 2) reduces the dimension of the *pdf* in the integrand.

2.1. Vecchia Approximation for the Gaussian pdf

Vecchia [14] introduced a way to approximate high-dimensional Gaussian *pdf*s arising from spatial data, which is particularly amenable to modification for our purposes. The starting point of the Vecchia [14] approximation is to write the joint density as a product of cascading conditional densities,

$$f(\mathbf{x}) = f(x_1) \prod_{i=2}^{D} f(x_i \,|\, \mathbf{x}_{1:i-1}). \tag{3}$$

Here, $f(x_1)$ is the univariate Gaussian density with mean μ_1 and variance Σ_{11}, and, for $i = 2, \ldots, k$, the conditional density $f(x_i \,|\, \mathbf{x}_{1:i-1})$ is the univariate Gaussian density with mean $\mu_i + \boldsymbol{\Sigma}_{[i,1:i-1]}\boldsymbol{\Sigma}^{-1}_{[1:i-1,1:i-1]}(\mathbf{x}_{1:i-1} - \boldsymbol{\mu}_{1:i-1})$ and variance $\Sigma_{i,i} - \boldsymbol{\Sigma}_{[i,1:i-1]}\boldsymbol{\Sigma}^{-1}_{[1:i-1,1:i-1]}\boldsymbol{\Sigma}_{[1:i-1,i]}$. The leading terms in this product are fast to calculate, but for terms corresponding to large i, the computations are nearly as burdensome as those of the original representation (1).

To help solve this problem, Vecchia [14] proposed an approximation to the full joint distribution, in the setting where the random vector $\mathbf{X}$ is observed from a spatial Gaussian process. He modified the cascading conditional representation (3) by replacing the conditioning on high-dimensional vectors $\mathbf{x}_{1:i-1}$ with conditioning on well-chosen vectors that have much smaller dimension. By limiting the conditioning sets to vectors of length $m << D$, this strategy replaces expensive $\mathcal{O}(D^3)$ matrix operations with much faster $\mathcal{O}(m^3)$ matrix operations. The approximation of the joint density is then

$$f(\mathbf{x}) \approx f(x_1) \prod_{i=2}^{D} f(x_i \,|\, \mathbf{x}_{\mathcal{N}_i}), \tag{4}$$

where $\mathcal{N}_i$ is the conditioning set of size m (more precisely, $\min(m, i-1)$) chosen for the component x_i, for $i = 2, \ldots, D$.

A good choice for a conditioning set to approximate the complete conditional density of each x_i might be the m components that are most correlated with x_i. In the context where the random vector $\mathbf{X} = (X(\mathbf{s}_1), \ldots, X(\mathbf{s}_k))^{\mathrm{T}}$ arises from a stationary spatial Gaussian process observed at locations $\mathbf{s}_1, \ldots, \mathbf{s}_k$, the components most correlated with $X_i \equiv X(\mathbf{s}_i)$ will be those observed at locations that are the m nearest neighbors to $\mathbf{s}_i$ (under covariance models in common use). Other strategies for constructing conditioning sets have also been explored [15,16].

Vecchia's approximation has been found to be quite accurate under many covariance models and sampling scenarios relevant to analysis of spatial Gaussian processes [16]. Moreover, it is very fast to compute, even using the most naive implementation. However, its power is fully realized when the D components of the product are computed in parallel, which is trivially easy to implement using standard tools in `R`.

2.2. Extending the Vecchia Approximation for the Gaussian cdf

Our approach to approximating the high-dimensional Gaussian *cdf* is to re-write the joint *cdf* as a telescoping product of conditional *cdf*s, analogously to (3), and then to approximate each complete conditional *cdf* with *cdf* that conditions on a smaller collection of components, analogously to (4). In the case of the *pdf*, this strategy of choosing smaller conditioning sets eliminates the need to compute high-dimensional matrix computations required by (1), whereas in the case of the *cdf*, this strategy eliminates the need to compute the high-dimensional integral required by (2).

Specifically, we can re-write any joint *cdf* as

$$\begin{aligned} F(\mathbf{x}) = P(\mathbf{X} < \mathbf{x}) &= P(X_1 < x_1) \prod_{i=2}^{D} P(X_i < x_i \,|\, X_1 < x_1, \ldots, X_{i-1} < x_{i-1}) \\ &= P(X_1 < x_1) \prod_{i=2}^{D} P(X_i < x_i \,|\, \mathbf{X}_{1:i-1} < \mathbf{x}_{1:i-1}) \end{aligned} \tag{5}$$

Then, just as in the approximation to the *pdf* (4), in the *cdf* (5) each conditional probability in the product can be approximated by reducing the size of the conditioning set to at most m components. Thus, our Vecchia approximation for the Gaussian *cdf* is

$$\begin{aligned} F(\mathbf{x}) &\approx P(X_1 < x_1) \prod_{i=2}^{D} P(X_i < x_i \,|\, \mathbf{X}_{\mathcal{N}_i} < \mathbf{x}_{\mathcal{N}_i}) \\ &= P(X_1 < x_1) \prod_{i=2}^{D} \frac{P(X_i < x_i, \mathbf{X}_{\mathcal{N}_i} < \mathbf{x}_{\mathcal{N}_i})}{P(\mathbf{X}_{\mathcal{N}_i} < \mathbf{x}_{\mathcal{N}_i})} \\ &= \Phi(x_1) \prod_{i=2}^{D} \frac{\Phi(\mathbf{x}_{\{i,\mathcal{N}_i\}})}{\Phi(\mathbf{x}_{\mathcal{N}_i})}, \end{aligned} \tag{6}$$

where again $\mathcal{N}_i$ is the conditioning set of size $\min(m, i-1)$ chosen for the component x_i, for $i = 2, \ldots, D$.

The approximation given by (6) reduces computational costs by replacing the D-dimensional integral in (2) with a series of much simpler integrals of dimension $m+1$

and m, for $m << D$. Furthermore, all of the elements in the product can be computed in parallel.

The multivariate *cdf*s in (6) still have to be evaluated numerically. For all but the smallest possible choices of m, best practices suggest using a QM method like that of Genz and Bretz [10] to approximate the numerator and denominator.

Similarly to the original Vecchia approximation to the Gaussian *pdf*, choosing the conditioning sets involves a trade-off; choose m too small and the accuracy of the approximation will suffer, but choose m too large and the computational benefits will diminish.

3. Simulation Study

To assess the accuracy and speed of this approximation, and to explore the trade-off inherent in the choice of m, we conduct a simulation study. Since the true value of the *cdf* is not available, the best we can do to check for accuracy is to see whether it is consistent with results obtained from direct use of the Genz and Bretz [10] QM approach. We simulate a Gaussian process observed on equally spaced grids of five different sizes, 15×15, 30×30, 50×50, 75×75 and 100×100. We try two different covariance functions for the Gaussian process to see whether this has an impact on the *cdf* estimation: an exponential model with range parameter 1 and and exponential model with range parameter 5, each with unit variance. This makes a total of 10 different scenarios. For each scenario, we used four different sizes of conditioning sets, choosing $m = 5$, 10, 30 and 50 closest neighbors. For comparison, we computed the Genz and Bretz [10] QM method using 499 and 3,607 sample points. We use the implementation of the Genz and Bretz [10] algorithm in the `mvPot` package [17] for `R`. In principle, the accuracy and computational requirements of the QM grows with the number of sample points (which, here, must be a prime number). Since the algorithms are stochastic, we repeated each calculation five times and plotted each replication as a dot in Figures 1, 2, 3, and 4.

Figure 1 shows the value of the estimated log *cdf* for all grid sizes and all estimation methods for the simulated Gaussian process with range parameter 1. The log *cdf* estimated with the Vecchia approximation increases with the number of neighbors until it stabilizes for 30 neighbors, after which it is consistent with the two QM approximations. This suggests that, under this scenario, it is advisable to use at least 30 neighbors in order to estimate the log *cdf*. For the two smaller grids, it appears that the Vecchia approximation has a similar variance to the QM approximation using 499 sample points, but a higher variance that the QM approximation using 3,607 sample points. For the larger grids, the Vecchia approximation appears to have a lower variance than both QM approximations. Figure 2 shows the same as Figure 1, but for exponential Gaussian processes with range parameter 5. The story is similar to the case with the shorter range process, except it appears that 50 neighbors may be necessary in order to stabilize the estimated log *cdf*. It may be the case that the number of neighbors necessary to accurately approximate the log *cdf* increases with length of the dependence of the Gaussian process. Intuitively, this may occur because for processes with longer-range dependence, a smaller proportion of the information in data may be captured by local approximations.

[Figure 1 about here.]

[Figure 2 about here.]

Figures 3 and 4 show the time required to approximate the log *cdf*, on a single core, for Gaussian processes with range parameters of 1 and 5, respectively. The computation time is influenced by both the number of observations and number of neighbors used in the Vecchia approximation. Computational costs increase with the number of observations, for both the Vecchia and QM approximation methods, and also increase with the number of neighbors in the conditioning set. Oddly, the empirical computation time did not increase for the QM approximation with the larger set of sample points. For smaller grid sizes, the QM methods are faster than the Vecchia approximations, except when the size of the conditioning set very small. For grids of size 50×50 and larger, computation time of the approximation using 30 neighbors was as fast as or faster than the QM method. When the number of observations is extremely large, in the case of the 100×100 grid, the computation time was much smaller for the Vecchia approximation compared to the QM approximation. This suggests that for high-dimensional datasets the use of the Vecchia approximation is preferable to the QM method, even if computations are done sequentially.

[Figure 3 about here.]

[Figure 4 about here.]

3.1. Parallel Computing

Since each term of the Vecchia *cdf* approximation (6) is independent of every term, it is trivial to parallelize the computations. In practice, we compute all of the required low-dimensional Gaussian *cdf*s on the log scale, and then sum them at the end. In principal, the speedup should be linear in the number of cores used for the calculation. To explore this relationship, we compute the *cdf* approximation based on a Gaussian process observed at 10,000 locations, varying the number of compute cores used between 5 and 40. For each setup, we repeat the computation 15 times. Figure 5 shows time required to compute the log *cdf* approximation. The computing time decreases with the number of cores. We observe roughly the expected linear relationship up to 20 cores, when a jump occurs before again decreasing. We suspect that this is behavior a result of the particular hardware configuration we used, which consists of networked 20-core processors. That is, we guess that once an additional physical processor is engaged, which occurs beyond 20 cores, overhead costs increase and attenuate the expected computational gains. When 40 cores were used, it took less than 1 minute to compute the log *cdf* approximation for 10,000 observations. There are clearly some diminishing returns due to communication overhead, but in principle, this approximation could be made arbitrarily fast with a big enough computing system.

[Figure 5 about here.]

3.2. Effect of Neighbor Selection and Joint Estimation

The representation defined by equation (5) and its approximation (6) calculates the joint probability as the product of univariate conditional distributions. However it is also possible to write the full joint *cdf* as a cascading product of multivariate, rather than univariate, conditional *cdf*s. Under equation 6, it is necessary to calculate the n univariate conditional probabilities, each of which requires a $m + 1$-dimensional *cdf* calculation. If instead we divide the components into q groups of p joint observations,

such that $q \times p = n$, we would only need to calculate the product of q conditional probabilities. However, doing so would make the dimensionality of each individual Gaussian *cdf* calculation in (6) between $m + p$ and $pm + p$. So it would trade the cost of computing higher-dimensional *cdf* terms for the benefit of computing fewer terms. Such a trade-off could affect both the accuracy and computational efficiency of the approximation. Guinness [16] explored this possibility in the context of *pdf*s and found that it can be advantageous to consider multivariate conditional densities in the Vecchia density approximation. To explore the effect of calculating higher dimensional conditional probabilities, we calculate the log *cdf* approximation based on groupings of observations of different sizes.

An additional consideration that could effect the accuracy and speed of the approximation is the construction of the conditioning sets. Using the nearest neighbors, as we have done above, requires the additional step of ordering the components by distance, which could be slow. Choosing randomly-selected conditioning sets could potentially speed up the computation by avoiding this sorting step.

Figures 6 and 7 show the estimated log *cdf* and time (in seconds) to compute the approximated log *cdf*, using the 100×100 grid. We used approximations based on joint conditional *cdf*s of dimension 2, 5, 10, 20, 30, and 50. For each grouping size, we constructed conditioning sets using 3 different methods. The first method conditions on the m most correlated observations (in this case simply the nearest neighbors) for each observation in the joint grouping, resulting in a conditioning set of size pm. The second method simply conditions on m random observations. The third method conditions on m random observations per element of the multivariate conditional calculation, again resulting of a conditioning set of size pm.

From Figure 6 it is clear that simply conditioning on m random observations fails to yield an acceptable approximation. Performance can be improved by conditioning on more random observations, which is what the third method does. Method 3 shows somewhat improved behavior, however it was only able to perform acceptably when both the dimensionality p of the joint conditional probability and the size of the conditioning set pm were both large. It is clear from Figure 6 that conditioning on random neighbors is much less accurate than conditioning on the most highly correlated neighbors. For conditioning sets consisting of small numbers m of neighbors per element in the joint conditional probability, the use of a large group p of joint observations had a better result, probably simply due to fact that the total number pm of neighbors in the conditioning set was larger. However, when the number m of correlated neighbors gets large enough the number p of joint observations does not seem to affect result of the approximation.

[Figure 6 about here.]

Figure 7 shows the computation time required for all of the approximation schemes depicted in Figure 6. The clear trend is that choosing a small conditioning set of random observations is very fast (middle panel), using higher-dimensional joint conditional *cdf*s is slower than using lower-dimensional joint conditional *cdf*s (all panels), and for the same size conditioning set, the time required to find the nearest neighbors is not a major bottleneck (right and left panels). This conclusion is different from exploration of the same issues, in the context of the *pdf*, found in Guinness [16]. There, using higher-dimensional joint conditional calculations was found to be beneficial, and the time required to find nearest neighbors was substantial enough to warrant the use of a fast approximate ordering algorithm. In the case of the *cdf* approximation, code

profiling confirmed that the time required to order the observations was insignificant, with the overwhelming majority of the computation time being used in calculating the lower-dimensional joint *cdf*s using the QM technique.

[Figure 7 about here.]

4. Example: A Gaussian Scale Mixture for Spatial Extremes

Recent advances in the statistics of extremal spatial phenomena have produced models that are flexible enough to accommodate both strong and weak spatial dependence in the far joint tails. One prominent strategy for achieving this is to construct scale mixtures of Gaussian processes, where the mixing distribution is chosen carefully so as to produce the desired tail dependence characteristics [4,6,18,19]. The preferred flavor of maximum likelihood inference for these models requires computing a Gaussian *cdf* whose dimension is roughly equal to the number of spatial locations in the dataset. Other state-of-the-art models for spatial extremes also rely on high-dimensional Gaussian *cdf*s [1–3]. To show the usefulness of our *cdf* approximation, we analyze data from precipitation gauges in Europe using the Gaussian scale mixture model from Huser et al. [6], which we describe below.

The class of scale mixtures of Gaussian processes is defined generically by

$$\begin{aligned} X(\mathbf{s}) &= R \times W(\mathbf{s}) \\ R &\sim F_R \perp\!\!\!\perp W(\mathbf{s}). \end{aligned} \tag{7}$$

Here, $W(\mathbf{s})$ is a standard Gaussian process (i.e. with unit variance) on some domain $\mathcal{D}$ indexed by $\mathbf{s} \in \mathcal{D}$. For a collection of k observations, the finite dimensional distribution of the Gaussian component is $\mathbf{W} \sim N_k(0, \mathbf{\Sigma}(\theta))$, where $\mathbf{\Sigma}(\theta)$ is a $D \times D$ covariance matrix constructed using a chosen covariance model that is indexed by parameter θ.

The random scaling R comes from distribution F_R. The choice of F_R is critical and determines the strength of the tail dependence in the resulting model [20]. A key quantity for summarizing the strength of tail dependence is the conditional probability $\chi_u(\mathbf{s}_i, \mathbf{s}_j) = P\{X(\mathbf{s}_i) > u \,|\, X(\mathbf{s}_j > u\}$, for spatial locations $\mathbf{s}_i$ and $\mathbf{s}_j$. If $\lim_{u\to\infty} \chi_u(\mathbf{s}_i, \mathbf{s}_j) = 0$ for all $\mathbf{s}_i, \mathbf{s}_j \in \mathcal{D}$, we say that $X(\mathbf{s})$ is *asymptotically independent*, while if $\lim_{u\to\infty} \chi_u(\mathbf{s}_i, \mathbf{s}_j) > 0$ for all $\mathbf{s}_i, \mathbf{s}_j \in \mathcal{D}$, we say that $X(\mathbf{s})$ is *asymptotically dependent*.

While many choices are available for the mixing distribution F_R, Huser et al. [6] suggest the parametric model defined by equation (8). When $\beta > 0$, the mixture process $X(\mathbf{s})$ is asymptotically independent, and when $\beta = 0$, $X(\mathbf{s})$ is asymptotically dependent. Therefore, this class of scale mixtures is rich enough to include both asymptotic independence and asymptotic dependence as nontrivial sub-models.

$$F_R(r) = \begin{cases} 1 - \exp\{-\gamma(r^\beta - 1)/\beta\}, & \text{for } \beta > 0 \\ 1 - r^\gamma, & \text{for } \beta = 0. \end{cases} \tag{8}$$

To construct the likelihood for maximum likelihood estimation, we must integrate out R from the model (7). Equations (9) and (10) show the marginal multivariate *cdf* and *pdf*, respectively, for a finite collection of observations $\mathbf{X}$ from $X(\mathbf{s})$ defined in (7). Here Φ_D represents the D-dimensional multivariate *cdf* from a Gaussian distribution

with mean vector 0 and covariance matrix $\mathbf{\Sigma}(\theta)$, and ϕ_D represents the D-dimensional multivariate *pdf* from a Gaussian distribution with mean 0 and covariance matrix $\mathbf{\Sigma}(\theta)$. There are no closed forms for these expressions, so it is necessary to use numerical methods to evaluate the (one-dimensional) integrals.

$$G(\mathbf{x}) = \int_0^\infty \Phi_D(\mathbf{x}/r; \mathbf{\Sigma}) f_R(r) dr \tag{9}$$

$$g(\mathbf{x}) = \int_0^\infty \phi_D(\mathbf{x}/r; \mathbf{\Sigma}) r^{-D} f_R(r) dr. \tag{10}$$

The preferred strategy for maximum likelihood estimation of extremal dependence models is to treat all observations falling below a high threshold as left censored [21]. This leads to a favorable balance between using the data as efficiently as possible, while not allowing data in the bulk of the distribution to have a large effect on dependence estimation. The censored likelihood for each temporal replicate is obtained by taking one partial derivative of (9) for every observation that falls above the threshold. Thus, (10) is the relevant likelihood when all observations, at one particular temporal replicate, are above the threshold, so nothing is censored. However, since the threshold is chosen to be a high quantile to prioritize inference on the tail, most observations are usually censored for any temporal replicate. When all observations fall below the threshold, the relevant likelihood is (9).

Most often, in any temporal replicate, there will be a mixture of observations above and below the threshold. In this case, the relevant joint likelihood of $\mathbf{x}$ is defined by equation (11), which results from taking partial derivatives of (9) with respect to only the un-censored observations. If we let I be the set of points above the threshold and I^c be the points below, then

$$\begin{aligned} G_I(\mathbf{x}) &:= \frac{\partial^{|I|}}{\partial \mathbf{x}_I} G(\mathbf{x}) = \int_0^\infty \frac{\partial^{|I|}}{\partial \mathbf{x}_I} \Phi_k(\mathbf{x}/r; \mathbf{\Sigma}) f_R(r) dr \\ &= \int_0^\infty \Phi_{|I^c|}\{(\mathbf{x}_{I^c} - \mathbf{\Sigma}_{I^c;I}\mathbf{\Sigma}_{I;I}^{-1}\mathbf{x}_I)/r; \mathbf{\Sigma}_{I^c|I}\} \phi_{|I|}(\mathbf{x}_I/r; \mathbf{\Sigma}_{I;I}) r^{-|I|} f_R(r) dr, \end{aligned} \tag{11}$$

where dependence of the covariance matrices on θ is suppressed for brevity, and the notation $\mathbf{\Sigma}_{A;A}$ refers to rows and columns of $\mathbf{\Sigma}$ pertinent to the points in A. The matrix $\mathbf{\Sigma}_{I^c|I} = \mathbf{\Sigma}_{I^c;I^c} - \mathbf{\Sigma} I^c; I\mathbf{\Sigma}_{I;I}^{-1}\mathbf{\Sigma}_{I;I^c}$ is the covariance matrix of the conditional normal distribution of the censored observations given the un-censored observations.

The computational issue arises because the integrand (11) contains a Gaussian *cdf* of dimension $|I^c|$, the number of censored observations in a temporal replicate. Again, for most replicates, this number $|I^c|$ is close to the total number of observation locations D because the censoring threshold is chosen to be high, such that most observations fall below the threshold and are therefore censored.

4.1. Precipitation Over Europe

Our dataset consists of weekly maximum precipitation observations between January, 2000 and April, 2019, in the western and central region of continental Europe, north of the mountain ranges the Pyrenees, Alps, and Carpathians. The 6 countries we consider

are Germany, Poland, Netherlands, Belgium, Czech Republic, and France. Figure 8 shows the locations of the observation stations distributed over Europe. This dataset consists of 1,006 weekly maxima from $D = 528$ weather stations. For context, the computational bottleneck from the Gaussian *cdf* limited the analysis in Huser et al. [6] to a dataset of $D = 12$ locations, even though analysis was performed on a large high-performance computing cluster. We use the weekly maximum daily accumulations at each location to break temporal dependence that might arise from storms that persist for more than one day. Out of the 531,168 total observations, 32.6% were missing values. For each weekly maximum, only the available data was used for estimation, and all missing observations were disregarded.

[Figure 8 about here.]

The covariance model we use for the underlying Gaussian processes is an anisotropic exponential, $\Sigma_{ij}(\theta) = \exp\{-h_{ij}/\rho\}$, where ρ is the range parameter and h_{ij} is the Mahalanobis distance between locations $\mathbf{s}_i$ and $\mathbf{s}_j$. The Mahalanois distance is parametrized as

$$h_{ij}^2 = \Omega^{\mathrm{T}}\Omega, \quad \text{where}\, \Omega = (\mathbf{s}_i - \mathbf{s}_j)^{\mathrm{T}} \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ 0 & A \end{pmatrix},$$

for rotation angle $\phi \in [0, \pi)$ and and aspect ratio $A > 1$. Thus, after fixing the mixing parameter γ at 1, as it plays a much less significant role than the parameter β in determining tail dependence characteristics, we arrive at a total of 4 parameters to estimate, $\psi = (\beta, \rho, \phi, A)^{\mathrm{T}}$.

The first step in estimating the dependence is to transform the observations to be on the same marginal scale. To do this, we start by applying a rank transformation to standard uniform, independently for each station. That is, for each station $k = 1, \ldots, D$ and each time point $t = 1, \ldots T$, the observation X_{kt} on the uniform scale is

$$U_{kt} = \frac{\mathrm{rank}(X_{kt})}{T+1}.$$

We next choose a high threshold to be the 0.95 marginal empirical quantile at each location. Then, denoting the marginal *cdf* and *pdf* of each X_{kt}, respectively, as $G_M(x) = \int_0^\infty \Phi(x/r) f(r) dr$ and $g_M(x) = \int_0^\infty \phi(x/r) r^{-1} f(r) dr$ (we assume stationarity, so the marginal distribution is assumed to be the same at each location), and letting the vector $\mathbf{v}_t = (\max\{u_{1t}, 0.95\}, \ldots, \max\{u_{Dt}, 0.95\})^{\mathrm{T}}$, the copula censored likelihood for each time replicate k is

$$L(\psi; \mathbf{v}_t) = \begin{cases} G\{G_M^{-1}(v_{1t}), \ldots, G_M^{-1}(v_{Dt})\} & \text{if all obs. are below the threshold} \\ \frac{g\{G_M^{-1}(v_{1t}), \ldots, G_M^{-1}(v_{Dt})\}}{\prod_{k=1}^{D} g_M\{G_M^{-1}(v_{kt})\}} & \text{if all obs. are above the threshold} \\ \frac{G_{I_t}\{G_M^{-1}(v_{1t}), \ldots, G_M^{-1}(v_{Dt})\}}{\prod_{k \in I_t}^{D} g_m\{G_M^{-1}(v_{kt})\}} & \text{if some obs. are above and some below the threshold} \end{cases}$$

Finally, the log likelihood across all time points t for the parameter vector ψ is

$$l(\psi; \mathbf{v}) = \sum_{t=1}^{T} \log(L(\psi; \mathbf{v}_t)).$$

We found the maximum likelihood estimator (MLE) by applying the Nelder-Mead numerical optimizer in the `R` function `optim`. MLEs are shown in Table 1. The MLE for the mixing parameter β is 0.82, which in this context is fairly far away from zero—far enough to strongly suggest that the process is asymptotically independent. The MLEs for the anisotropy parameters suggest pronounced eccentricity. To interpret and visualize the estimated dependence model implied by the MLEs shown in Table 1, we plot level curves in the resulting χ_u function for $u = 0.95$ on the quantile scale, shown in Figure 8. Each ellipse represents a constant value of $\chi_{u=0.95}(\mathbf{s}) = P\{F_M[X(\mathbf{s})] > 0.95 \,|\, F_M[X(\mathbf{s}_0)] > 0.95\}$, for an arbitrarily-chosen reference point $\mathbf{s}_0$ near the center of the map. The level curves are ellipses due to the anisotropic construction, with the major axis roughly along a northeast-southwest orientation, and joint exceedances more likely with decreasing distance from $\mathbf{s}_0$.

[Table 1 about here.]

5. Discussion

The main objective of this paper was to propose fast approximation to high-dimensional Gaussian *cdf*s that arise from spatial Gaussian processes. We modified Vecchia's approximation for Gaussian *pdf*s to the context of Gaussian *cdf*s. Simulations showed that for large numbers of locations and relatively small conditioning sets, this approximation gives results consistent with state-of-the-art QM methods, and reduces computational time considerably, even when computations are performed sequentially. Furthermore, the approximation is trivially easy to code in parallel using standard `R` packages, and requires no linking to specialized software libraries.

We demonstrated the utility of our fast *cdf* approximation by using it to find maximum censored likelihood estimates for the scale mixture model of Huser et al. [6]. This model is attractive because of its flexible tail dependence characteristics, but is hampered by computational difficulties arising from the need to compute high-dimensional Gaussian *cdf*s during inference. We fit this model to a precipitation dataset consisting over 500 spatial locations, whereas previous efforts using conventional QM techniques were limited to just 12 locations.

One drawback that we noticed during the data analysis is that conventional optimization routines had trouble converging, due to the stochastic nature of the likelihood objective function. For future studies, one possible approach to circumventing this problem is to use stochastic optimization algorithms, which may be better suited to optimizing random objective functions.

Code to reproduce the simulations can be found in our Git repository. `https://github.com/Recca2012/CDFApprox.git`

Acknowledgements

This research was supported in part by NSF grant DMS-1752280. Computations for this research were performed on the Institute for Computational and Data Sciences Advanced CyberInfrastructure (ICDS-ACI).

References

[1] J. L. Wadsworth, J. A. Tawn, Efficient inference for spatial extreme value processes associated to log-Gaussian random functions, Biometrika 101 (1) (2014) 1–15, ISSN 0006-3444, , URL https://doi.org/10.1093/biomet/ast042.

[2] E. Thibaud, J. Aalto, D. S. Cooley, A. C. Davison, J. Heikkinen, Bayesian inference for the Brown-Resnick process, with an application to extreme low temperatures, Ann. Appl. Stat. 10 (4) (2016) 2303–2324, ISSN 1932-6157, , URL https://doi.org/10.1214/16-AOAS980.

[3] R. de Fondeville, A. C. Davison, High-dimensional peaks-over-threshold inference, Biometrika 105 (3) (2018) 575–592, ISSN 0006-3444, , URL https://doi.org/10.1093/biomet/asy026.

[4] R. Huser, J. L. Wadsworth, Modeling spatial processes with unknown extremal dependence class, J. Amer. Statist. Assoc. 114 (525) (2019) 434–444, ISSN 0162-1459, , URL https://doi.org/10.1080/01621459.2017.1411813.

[5] R. B. Arellano-Valle, A. Azzalini, On the unification of families of skew-normal distributions, Scand. J. Statist. 33 (3) (2006) 561–574, ISSN 0303-6898, , URL https://doi.org/10.1111/j.1467-9469.2006.00503.x.

[6] R. Huser, T. Opitz, E. Thibaud, Bridging asymptotic independence and dependence in spatial extremes using Gaussian scale mixtures, Spat. Stat. 21 (part A) (2017) 166–186, , URL https://doi.org/10.1016/j.spasta.2017.06.004.

[7] A. Genz, Numerical Computation of Multivariate Normal Probabilities, Journal of Computational and Graphical Statistics 1 (2) (1992) 141–149.

[8] A. Genz, Numerical computation of rectangular bivariate and trivariate normal and t probabilities, Stat. Comput. 14 (3) (2004) 251–260, ISSN 0960-3174, , URL https://doi.org/10.1023/B:STCO.0000035304.20635.31.

[9] T. Miwa, A. J. Hayter, S. Kuriki, The evaluation of general non-centred orthant probabilities, J. R. Stat. Soc. Ser. B Stat. Methodol. 65 (1) (2003) 223–234, ISSN 1369-7412, , URL https://doi.org/10.1111/1467-9868.00382.

[10] A. Genz, F. Bretz, Computation of multivariate normal and t probabilities, vol. 195 of *Lecture Notes in Statistics*, Springer, Dordrecht, ISBN 978-3-642-01688-2, , URL https://doi.org/10.1007/978-3-642-01689-9, 2009.

[11] M. G. Genton, D. E. Keyes, G. Turkiyyah, Hierarchical decompositions for the computation of high-dimensional multivariate normal probabilities, J. Comput. Graph. Statist. 27 (2) (2018) 268–277, ISSN 1061-8600, , URL https://doi.org/10.1080/10618600.2017.1375936.

[12] W. Hackbusch, Hierarchical matrices: algorithms and analysis, vol. 49 of *Springer Series in Computational Mathematics*, Springer, Heidelberg, ISBN 978-3-662-47323-8; 978-3-662-47324-5, , URL https://doi.org/10.1007/978-3-662-47324-5, 2015.

[13] J. Cao, M. G. Genton, D. E. Keyes, G. M. Turkiyyah, Hierarchical-block conditioning approximations for high-dimensional multivariate normal probabilities, Stat. Comput. 29 (3) (2019) 585–598, ISSN 0960-3174, , URL https://doi.org/10.1007/s11222-018-9825-3.

[14] A. V. Vecchia, Estimation and Model Identification for Continuous Spatial Processes, Journal of the Royal Statistical Society. Series B (Methodological) 50 (2) (1988) 297–312.

[15] M. L. Stein, Z. Chi, L. J. Welty, Approximating likelihoods for large spatial data sets, J. R. Stat. Soc. Ser. B Stat. Methodol. 66 (2) (2004) 275–296, ISSN 1369-7412.

[16] J. Guinness, Permutation and grouping methods for sharpening Gaussian process approximations, Technometrics 60 (4) (2018) 415–429, ISSN 0040-1706, , URL https://doi.org/10.1080/00401706.2018.1437476.

[17] R. de Fondeville, L. Belzile, mvPot: Multivariate Peaks-over-Threshold Modelling for Spatial Extreme Events, URL https://CRAN.R-project.org/package=mvPot, r package version 0.1.4, 2018.

[18] T. Opitz, Modeling asymptotically independent spatial extremes based on Laplace random

fields, Spat. Stat. 16 (2016) 1–18, , URL https://doi.org/10.1016/j.spasta.2016.01.001.

[19] S. A. Morris, B. J. Riech, E. Thibaud, D. Cooley, A space-time skew-t model for threshold exceedances, Biometrics 73 (3) (2017) 749–758, ISSN 0006-341X, , URL https://doi.org/10.1111/biom.12644.

[20] S. Engelke, T. Opitz, J. Wadsworth, Extremal dependence of random scale constructions, Extremes 22 (4) (2019) 623–666, ISSN 1386-1999, , URL https://doi.org/10.1007/s10687-019-00353-3.

[21] R. Huser, A. C. Davison, M. G. Genton, Likelihood estimators for multivariate extremes, Extremes 19 (1) (2016) 79–103, ISSN 1386-1999, , URL https://doi.org/10.1007/s10687-015-0230-4.

Parameter	MLE
ρ	1.31
β	0.82
ϕ	1.10
A	2.29

Table 1. Maximum likelihood estimates of dependence parameters

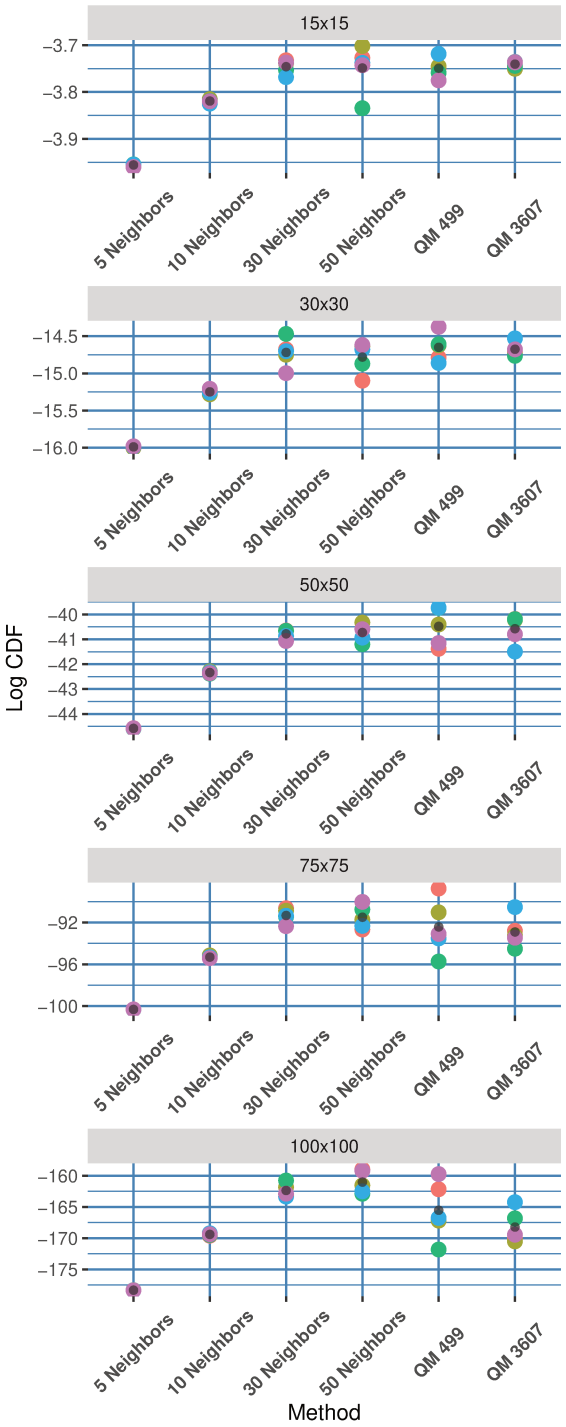

Figure 1. Estimated log *cdf* for exponential Gaussian processes with range parameter $\rho = 1$. The x-axis represents the different methods used for the *cdf* computation and the y-axis is the log *cdf*. Each point is an independent estimate of the log *cdf*, and each black point is the average over the replications. The Vecchia approximation seems to stabilize when at least 30 neighbors are used, and results in values that are consistent with the QM approximations.

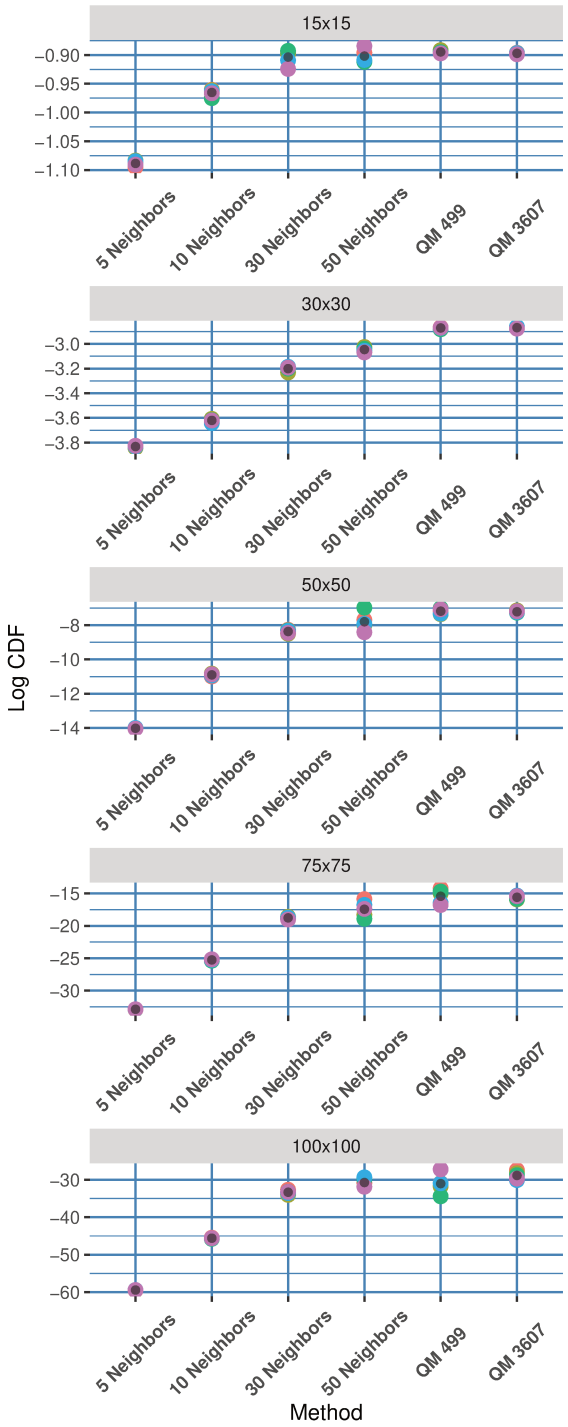

Figure 2. Estimated log *cdf* for exponential Gaussian processes with range parameter $\rho = 5$. The x-axis represents the different methods used for the *cdf* computation and the y-axis is the log *cdf*. Each point is an independent estimate of the log *cdf*, and each black point is the average over the replications. For this process with longer-range dependence, the Vecchia approximation may not stabilize until at least 50 neighbors are used, when results become consistent with the QM approximations.

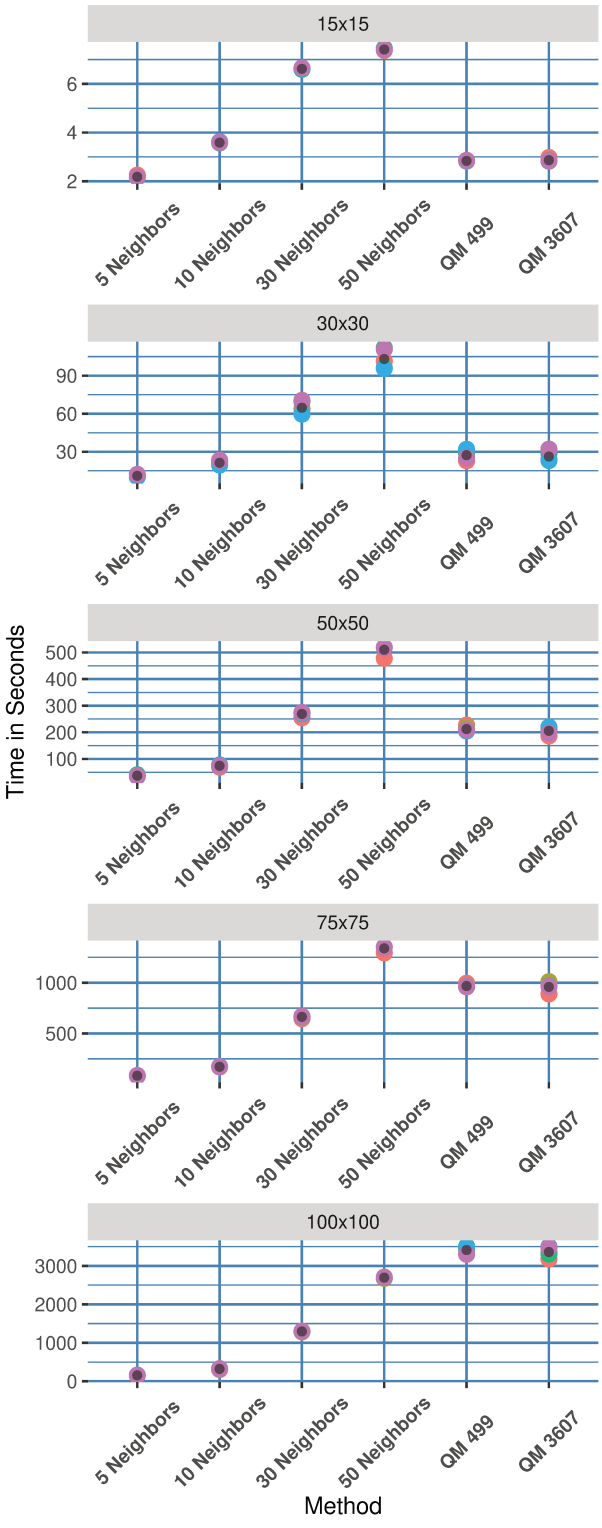

Figure 3. Time to estimate the *cdf* approximation for an exponential Gaussian process with range parameter $\rho = 1$. The x-axis represents the different approximation methods, and the y-axis is the computation time. Each point is an independent replication of the procedure, and the black point is the average over the replications.

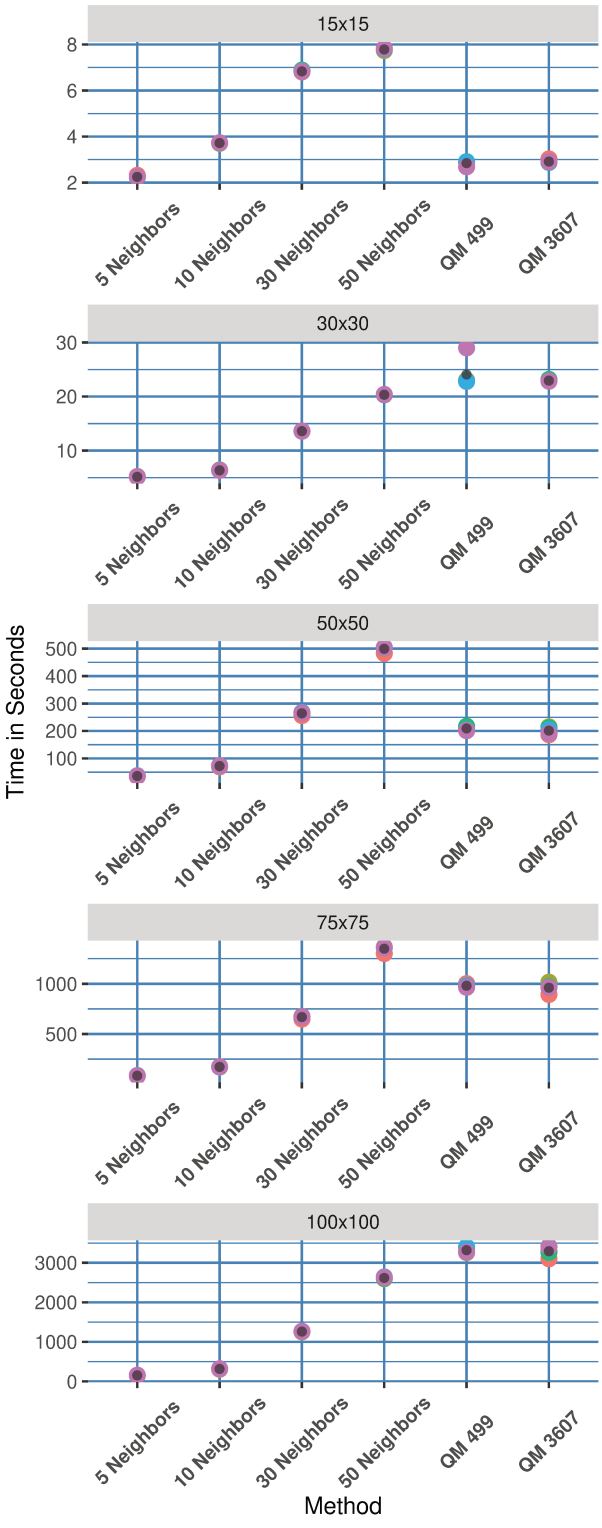

Figure 4. Time to estimate the *cdf* approximation for an exponential Gaussian process with range parameter $\rho = 5$. The x-axis represents the different approximation methods, and the y-axis is the computation time. Each point is an independent replication of the procedure, and the black point is the average over the replications.

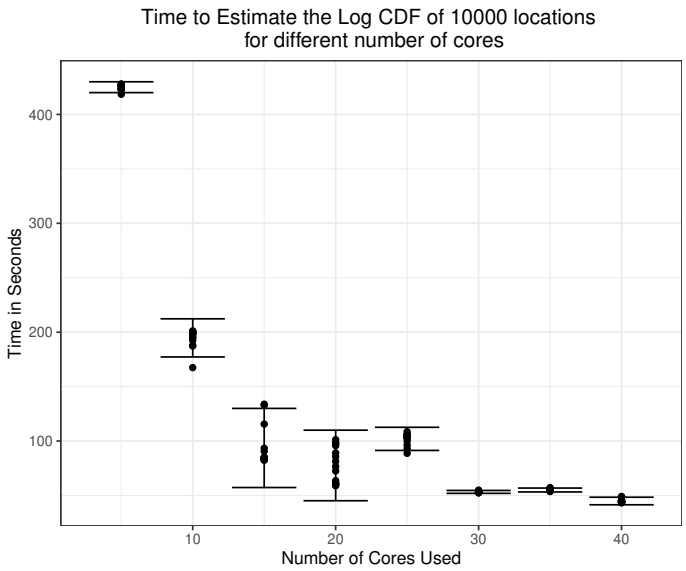

Figure 5. Time to compute log *cdf* approximation parallelized across different numbers of computing cores.

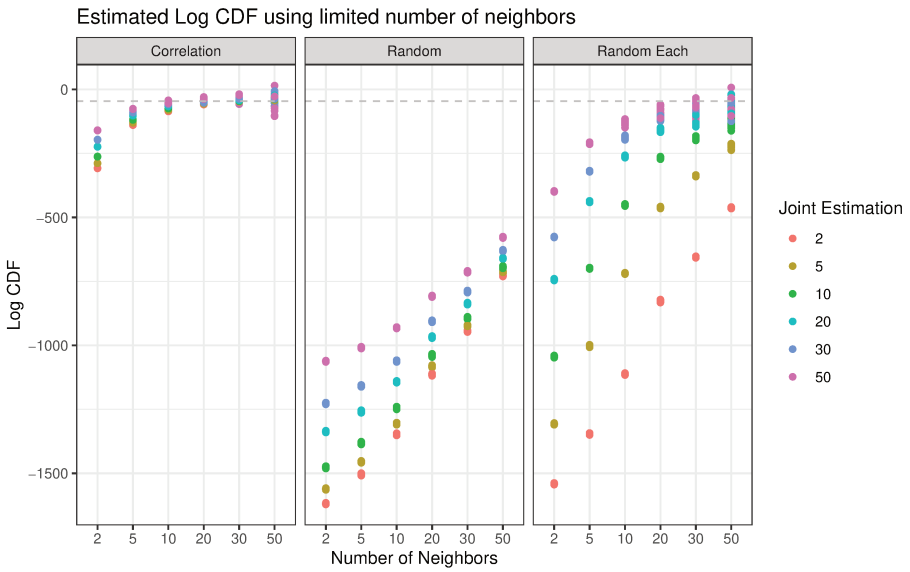

Figure 6. Estimated log *cdf* based on observations from an exponential Gaussian process with range parameter $\rho = 10$, using 3 different methods to select conditioning sets, and different dimensionalities of joint conditional observations. Since the true value of the *cdf* is unknown, we depict the mean value of five QM estimates, each using 3607 sample points, as a dashed line, representing the best available estimate of the true value.

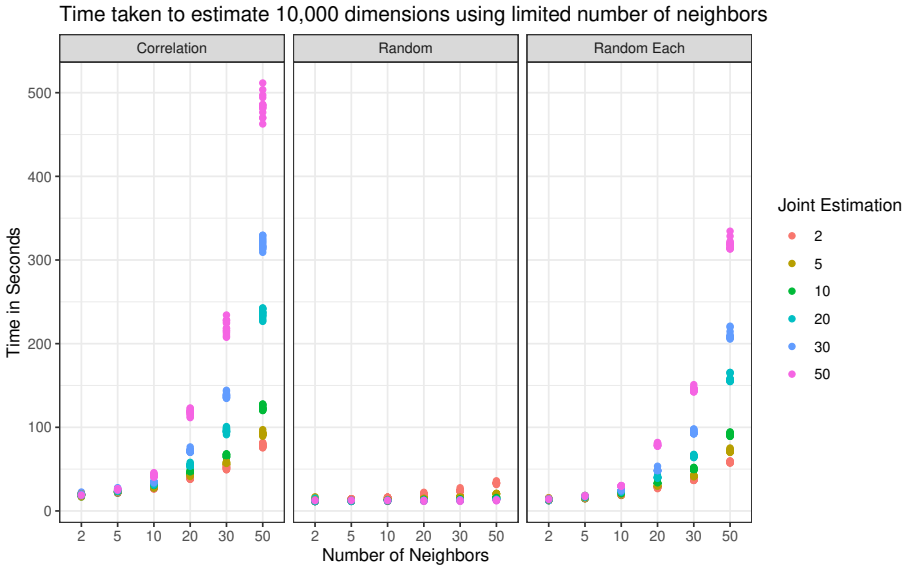

Figure 7. Time to estimate the log CDF with dependence parameter $\rho = 1$ using 3 different methods to select neighbors, multiple number of neighbors and multiple number of joint observations.

Figure 8. $D = 528$ weather stations located over 6 European countries