Published in final edited form as:
Commun Stat Simul Comput. 2020 ; 49(8): 1957–1981. doi:10.1080/03610918.2018.1508705.

MAXIMUM LIKELIHOOD ESTIMATION OF GAUSSIAN COPULA MODELS FOR GEOSTATISTICAL COUNT DATA

Zifei Han,
Vertex Pharmaceuticals, Boston MA 02210, USA, hanzifei1@gmail.com

Victor De Oliveira
Department of Management Science and Statistics, The University of Texas at San Antonio, San Antonio, TX 78249, USA, victor.deoliveira@utsa.edu

Abstract

This work investigates the computation of maximum likelihood estimators in Gaussian copula models for geostatistical count data. This is a computationally challenging task because the likelihood function is only expressible as a high dimensional multivariate normal integral. Two previously proposed Monte Carlo methods are reviewed, the Genz–Bretz and Geweke–Hajivassiliou–Keane simulators, and a new method is investigated. The new method is based on the so–called *data cloning* algorithm, which uses Markov chain Monte Carlo algorithms to approximate maximum likelihood estimators and their (asymptotic) variances in models with computationally challenging likelihoods. A simulation study is carried out to compare the statistical and computational efficiencies of the three methods. It is found that the three methods have similar statistical properties, but the Geweke–Hajivassiliou–Keane simulator requires the least computational effort. Hence, this is the method we recommend. A data analysis of Lansing Woods tree counts is used to illustrate the methods.

Keywords

Data cloning; Gaussian random field; Markov chain Monte Carlo; Multivariate normal integral; Simulated likelihood

AMS subject classifications:

60G10; 60G60; 62M30

1 Introduction

Geostatistical data are a kind of spatial data collected in many earth and social sciences such as ecology, demography and geology. The literature offers a large number of models for the analysis of *continuous* geostatistical data but, in comparison, the number of models for the analysis of *count* geostatistical data is rather limited. Two main classes of models have been proposed for the analysis of the latter, both of which use Gaussian random fields as building blocks. The first is the class of hierarchical models proposed by Diggle et al. (1998), which can also be viewed as a class of generalized linear mixed models. De Oliveira (2013) proposed a class of hierarchical Poisson models that generalize the Poisson–lognormal model used by Diggle et al. (1998), and studied the second–order properties of these models. It was shown in De Oliveira (2013) that these models lack the flexibility to describe moderate or strong correlation in datasets that consist mostly of small counts or have small overdispersion. The second is the class of Gaussian copula models proposed by Madsen (2009), Kazianka and Pilz (2010) and Kazianka (2013). Han and De Oliveira (2016) studied the correlation structure of geostatistical Gaussian copula models with two types of marginal distributions, and found that their correlation structure is more flexible than that of hierarchical Poisson models. The latter is the class of models to be considered in this work.

Likelihood–based inference for Gaussian copula models with discrete marginals is computationally challenging, because evaluation of the likelihood functions require approximating high dimensional multivariate normal integrals. To avoid this computational challenge, Kazianka and Pilz (2010), Kazianka (2013) and Bai et al. (2014) proposed basing inference on some form of pseudo–likelihood that has similar properties as a true likelihood, but takes less time to compute. These works proposed the use of pairwise likelihoods and generalized quantile transforms. The former uses bivariate copulas, which is an example of the composite likelihood approach (see Varin et al. (2011) for a thorough review), while the latter uses a continuous approximation of the discrete distribution, which is based on copula densities. The statistical efficiencies of these approaches were numerically explored by Kazianka (2013) and Nikoloulopoulos (2013, 2016). These authors found that, compared to maximum likelihood estimators, the time savings in the computation of these pseudo–likelihood based estimators are obtained at the expense of a substantial increase of their biases and variances. Hence maximum likelihood estimators are preferable in situations in which they can be computed in a reasonable amount of time. In this work we study methods for efficient evaluation of likelihood functions of Gaussian copula models with count marginal distributions.

The goals of this work are threefold. First, we review two *simulated likelihood* methods that have been proposed in the literature for the approximation of high dimensional multivariate normal integrals using Monte Carlo methods. One of them is the quasi–Monte Carlo method proposed by Genz (1992) and Genz and Bretz (2002), while the other is the Geweke–Hajivassiliou–Keane simulator proposed by Geweke (1991), Hajivassiliou et al. (1996) and Keane (1994). Nikoloulopoulos (2013, 2016) and Masarotto and Varin (2012) used these methods to approximate likelihoods of Gaussian copula models with count marginals, which were then used to numerically compute the maximum simulated likelihood estimators. Second, we investigate the use of *data cloning* to compute maximum likelihood estimators (MLE). Data cloning is a method proposed by Lele et al. (2007) to approximate MLE and their (asymptotic) variances using Markov chain Monte Carlo methods. Finally, we carry out a simulation study to compare the statistical and computational efficiencies of these three methods, and compare the behavior of three types of confidence intervals for the model parameters. The simulated likelihood methods are implemented in the R packages mvtnorm (Genz and Bretz, 2009), gcmr (Masarotto and Varin, 2017) and gcKrig (Han and De

Oliveira, 2018), while the data cloning method was coded and implemented by us in R and C++.

Similar simulation studies were carried out by Nikoloulopoulos (2013, 2016), in which the quasi–Monte Carlo, simple Monte Carlo and the pseudo–likelihood methods were compared. In those, the quasi–Monte Carlo method was the most effective regarding bias and variance, but the computational effort was *not* considered. Our study, tailored to the geostatistical context, compares methods not considered by Nikoloulopoulos, includes larger dimensional settings and compares both statistical and computational efficiency. The results of this simulation study indicate that, on the one hand, the three methods to approximate MLE have similar statistical properties. On the other hand, the quasi–Monte Carlo and data cloning methods require the most computational effort, while the Geweke–Hajivassiliou–Keane simulator requires substantially least computational effort. So the latter provides the best balance between statistical and computational efficiency. An analysis of the Lansing Woods tree counts dataset is used to illustrate the application of the methods for several inferential tasks.

2 Gaussian Copula Models with Discrete Marginals

2.1 Model Description

Let $\{Y(\mathbf{s}) : \mathbf{s} \in D\}, D \subset \mathbb{R}^2$, be a random field taking values on $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$, and let $\{F_{\mathbf{s}}(\cdot\,; \psi) : \mathbf{s} \in D\}$ be a family of marginal cdfs with support contained in $\mathbb{N}_0$ and corresponding pmfs $f_{\mathbf{s}}(\cdot\,; \psi)$, where ψ are *marginal parameters*. It is assumed that $F_{\mathbf{s}}(\cdot\,; \psi)$ is parameterized by $\psi = (\beta^\top, \sigma^2)$, in a way that

$$\mathrm{E}(Y(\mathbf{s})) = t(\mathbf{s}) \exp(\boldsymbol{\beta}^\top \mathbf{f}(\mathrm{s})),$$

where $\beta = (\beta_1, \ldots, \beta_p)^\top$ are regression parameters, $\mathbf{f}(\mathbf{s}) = (f_1(\mathbf{s}), \ldots, f_p(\mathbf{s}))^\top$ are known location–dependent covariates, and $t(\mathbf{s})$ is a known ‘sampling effort’. The parameter $\sigma^2 > 0$ controls dispersion (usually overdispersion). For instance, when the support of the cdfs is $\mathbb{N}_0$, σ^2 may control the departure of $F_{\mathbf{s}}(\cdot\,; \psi)$ from the Poisson distribution with the same mean. The negative binomial and zero–inflated Poisson distributions are two common examples.

The Gaussian copula model for $Y(\cdot)$ is the random field defined by the property that for every $m \in \mathbb{N}$ and $\mathbf{s}_1, \ldots, \mathbf{s}_m \in D$, the joint cdf of $(Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_m))$ is

$$P(Y(\mathbf{s}_1) \le y_1, \ldots, Y(\mathbf{s}_m) \le y_m) = \Phi_m\left(\Phi^{-1}(F_{\mathbf{s}_1}(y_1; \boldsymbol{\psi})), \ldots, \Phi^{-1}(F_{\mathbf{s}_m}(y_m; \boldsymbol{\psi})); \Psi_{\boldsymbol{\vartheta}}\right), \qquad (1)$$

where $\Phi(\cdot)$ is the cdf of the standard normal distribution and $\Phi_m(\cdot\,; \Psi_{\vartheta})$ is the cdf of the $\mathrm{N}_m(\mathbf{0}, \Psi_{\vartheta})$ distribution. The (i, j)th entry of the $m \times m$ correlation matrix Ψ_{ϑ} is assumed to be

$$K_{\boldsymbol{\vartheta}}(\mathbf{s}_i, \mathbf{s}_j) = (1 - \tau^2)\bar{K}_\theta(\mathbf{s}_i, \mathbf{s}_j) + \tau^2 \mathbf{1}\{\mathbf{s}_i = \mathbf{s}_j\}, \qquad (2)$$

where $\bar{K}_\theta(\mathbf{s}_i, \mathbf{s}_j) \geq 0$ is a correlation function in $\mathbb{R}^2$ that is continuous everywhere, $\boldsymbol{\vartheta} = (\boldsymbol{\theta}^\top, \tau^2)$ are correlation parameters, with $\tau^2 \in [0, 1]$, and $\mathbf{1}\{A\}$ is the indicator function of A. This includes both correlation functions that are continuous everywhere and correlation functions that are discontinuous along the 'diagonal' $\mathbf{s}_i = \mathbf{s}_j$.

2.2 Likelihood

The computation of the likelihood function of Gaussian copulas with discrete marginals is challenging because it lacks a closed–form expression (unlike the case with continuous marginals), so numerical computation is required. Here we review several representations of this likelihood that have appeared in the literature.

Suppose that $Y(\cdot)$ is observed at n distinct sampling locations $\mathbf{s}_1, \ldots, \mathbf{s}_n \in D$, resulting in the counts $\boldsymbol{y} = (y_1, \ldots, y_n)$, where y_i is the realization of $Y(\mathbf{s}_i)$. Then the likelihood function of $\boldsymbol{\eta} = (\boldsymbol{\psi}^\top, \boldsymbol{\vartheta}^\top)$ based on the data $\mathbf{y}$ is given by

$$\begin{aligned} L(\boldsymbol{\eta}; \boldsymbol{y}) &= \mathrm{P}_\eta(Y(\mathbf{s}_1) = y_1, \ldots, Y(\mathbf{s}_n) = y_n) \\ &= \sum_{j_1=1}^{2} \cdots \sum_{j_n=1}^{2} (-1)^{\sum_{i=1}^n j_i} \Phi_n\Big(\Phi^{-1}(F_{\mathbf{s}_1}(t_1; \boldsymbol{\psi})), \ldots, \Phi^{-1}(F_{\mathbf{s}_n}(t_n; \boldsymbol{\psi})); \Psi_{\boldsymbol{\vartheta}}\Big), \end{aligned} \qquad (3)$$

with $t_i = y_i + 1 - j_i$, where the latter equality follows by taking the Radon–Nikodym derivative of (1) with respect to the counting measure on $\mathbb{N}_0^n$ (Song, 2000). Evaluation of (3) is computationally intractable, even for moderate n, since it requires computing 2^n terms, each involving an n–dimensional integral.

Alternatively, note that

$$Y(\mathbf{s}) \overset{d}{=} F_{\mathbf{s}}^{-1}(\Phi(Z(\mathbf{s})); \boldsymbol{\psi}), \qquad (4)$$

where $F_{\mathbf{s}}^{-1}(u; \boldsymbol{\psi}) = \inf\{x \in \mathbb{R} : F_{\mathbf{s}}(x; \boldsymbol{\psi}) \geq u\}$, $u \in (0, 1)$, is the quantile function of $F_{\mathbf{s}}(\cdot\,; \boldsymbol{\psi})$, and $Z(\cdot)$ is the Gaussian random field with mean 0 and covariance function (2) (Han and De Oliveira, 2016). For count cdfs, $F_{\mathbf{s}}^{-1}(\cdot\,; \boldsymbol{\psi})$ is a step function so for any $\mathbf{s} \in D$ and $y \in \mathbb{N}_0$ it holds that $Y(\mathbf{s}) = y$ if and only if $F_{\mathbf{s}}(y - 1; \boldsymbol{\psi}) < \Phi(Z(\mathbf{s})) \leq F_{\mathbf{s}}(y; \boldsymbol{\psi})$. Then the likelihood function can also be written as

$$\begin{aligned} L(\boldsymbol{\eta}; \boldsymbol{y}) &= \mathrm{P}_\eta(\Phi^{-1}(F_{\mathbf{s}_i}(y_i - 1; \boldsymbol{\psi})) < Z(\mathbf{s}_i) \leq \Phi^{-1}(F_{\mathbf{s}_i}(y_i; \boldsymbol{\psi})) : i = 1, \ldots, n) \\ &= \int_{\Phi^{-1}(F_{\mathbf{s}_1}(y_1 - 1; \boldsymbol{\psi}))}^{\Phi^{-1}(F_{\mathbf{s}_1}(y_1; \boldsymbol{\psi}))} \cdots \int_{\Phi^{-1}(F_{\mathbf{s}_n}(y_n - 1; \boldsymbol{\psi}))}^{\Phi^{-1}(F_{\mathbf{s}_n}(y_n - 1; \boldsymbol{\psi}))} \phi_n(z_1, \ldots, z_n; \Psi_{\boldsymbol{\vartheta}}) dz_1, \ldots dz_n, \end{aligned} \qquad (5)$$

where $\varphi_n(\cdot\,; \Psi_{\boldsymbol{\vartheta}})$ is the pdf of the $\mathrm{N}_n(\mathbf{0}, \Psi_{\boldsymbol{\vartheta}})$ distribution. The latter expression is much more amenable for numerical computation, and will be used in the next section as the starting point for likelihood approximation.

Yet another likelihood representation is given in Masarotto and Varin (2012). Consider the change of variables $(z_1, \ldots, z_n) \to (u_1, \ldots, u_n)$ given by

$$u_i = \frac{F_{\mathbf{s}_i}(y_i; \boldsymbol{\psi}) - \Phi(z_i)}{f_{\mathbf{s}_i}(y_i; \boldsymbol{\psi})}, \qquad i = 1, \ldots, n, \tag{6}$$

with inverse $z_i = z_i(u_i) = \Phi^{-1}\,(F_{\mathbf{s}_i}(y_i;\ \psi) - u_i f_{\mathbf{s}_i}(y_i;\ \psi))$. Then the integral (5) can be rewritten as

$$L(\boldsymbol{\eta}; \boldsymbol{y}) = \prod_{i=1}^{n} f_{\mathbf{s}_i}(y_i; \boldsymbol{\psi}) \cdot \int_0^1 \cdots \int_0^1 |\,\Psi_{\vartheta}\,|^{-\frac{1}{2}} \exp\!\left(-\frac{1}{2}\boldsymbol{z}(\mathbf{u})^{\top}(\Psi_{\vartheta}^{-1} - I_n)\boldsymbol{z}(\mathbf{u})\right) d\mathbf{u}, \tag{7}$$

where $\boldsymbol{z}(\mathbf{u}) = (z_1(u_1), \ldots, z_n(u_n))^{\top}$; see Appendix A for the proof. It is also shown in Appendix A that the right-hand side of (7) agrees with the likelihood representation obtained by Madsen and Fang (2011) using a ‘continuous extension’ argument.

Madsen and Fang (2011) approximated the right–hand side of (7) using simple Monte Carlo based on independent draws from the unif(0, 1) distribution, but this results in an inefficient algorithm, even when the dimension n is small (Nikoloulopoulos, 2013). Approximating the right–hand side of (5) using the simple Monte Carlo algorithm based on independent draws from the $N_n(\mathbf{0}, \Psi_{\vartheta})$ distribution would also be inefficient, so more efficient algorithms are described in the next section.

Remark 1. In the statistical literature about copulas it is well known the lack of uniqueness of copula representations for multivariate distributions with discrete marginals. We do not view this as a weakness of copulas models for discrete spatial data though, as long as the copula model that is used is flexible and fits well the data under study; this is illustrated in Section 7. In addition, it is worth noting that the aforementioned lack of uniqueness does not arise within the class of Gaussian copula models because for any n and $u_1, \ldots, u_n$, the copula $\Phi_n(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_n); \Psi_{\vartheta})$ is a one–to–one function of Ψ_{ϑ}.

3 Simulated Likelihood Methods

Approximating the n–dimensional integral in (5) by (deterministic) quadrature methods is not practical for moderate or large n, due to the high computational cost and large errors involved, so Monte Carlo or quasi–Monte Carlo methods are preferable. We describe in this section two such methods that have been proposed in the literature. Then the (simulated) maximum likelihood estimates would be computed by (numerically) maximizing an approximation to the likelihood that is obtained by simulation.

3.1 Genz–Bretz Method

We briefly summarize the method proposed by Genz (1992) and Genz and Bretz (2002) to compute n–dimensional integrals of normal densities over bounded hyper–rectangles; it will be called here the GB method. First, a three–step transformation (change of variable) is applied to the integrand, aimed at transforming (5) into an integral where the integration region is the hypercube $[0, 1]^{n-1}$. To simplify the expressions that follow, let $a_i = \Phi^{-1}(F_{\mathbf{s}_i}(y_i - 1;\ \psi))$ and $b_i = \Phi^{-1}(F_{s_i}(y_i;\ \psi))$. After such transformation the likelihood (5) is rewritten as

$$L(\boldsymbol{\eta}; \boldsymbol{y}) = \int_0^1 \cdots \int_0^1 \prod_{i=1}^{n} (e_i - d_i)\, dw_1 \dots dw_{n-1}, \tag{8}$$

where $e_i = \Phi((b_i - \sum_{j=1}^{i-1} c_{ij}\Phi^{-1}(d_j + w_j(e_j - d_j))) / c_{ii})$, $d_i = \Phi((a_i - \sum_{j=1}^{i-1} c_{ij}\Phi^{-1}(d_j + w_j(e_j - d_j))) / c_{ii})$ and[1] $(c_{jk})_{1 \le k \le j \le n}$ are the entries of the lower triangular matrix in the Cholesky decomposition of Ψ_{ϑ}. Note that the dimension of the integral was reduced by one. Second, a randomized quasi–Monte Carlo method with antithetic variables is used. Given the Monte Carlo size M and number of quasi–random points P, the GB method approximates (8) by

$$\hat{L}^{\mathrm{GB}}(\boldsymbol{\eta}; \boldsymbol{y}) = \frac{1}{M}\sum_{k=1}^{M} \frac{1}{2P}\sum_{j=1}^{P} (f(\,|\,2\{\boldsymbol{p}_j + \boldsymbol{W}_k\} - \mathbf{1}_{n-1}\,|\,) + f(\mathbf{1}_{n-1} - \,|\,2\{\boldsymbol{p}_j + \boldsymbol{W}_k\} - \mathbf{1}_{n-1}\,|\,)),$$

where $f(\cdot)$ is the integrand in (8), $\{\boldsymbol{t}\}$ and $|\boldsymbol{t}|$ denote, respectively, the vectors obtained by taking the fractional part and absolute value of each component of $\boldsymbol{t}$, and $\mathbf{1}_{n-1}$ is the vector of ones. For $k = 1, \dots, M$ and $j = 1, \dots, P$, $\boldsymbol{W}_k$ is an $(n - 1)$–vector whose components are independent draws from the unif(0, 1) distribution, and $\boldsymbol{p}_j$ is an $(n - 1)$–vector whose components are quasi–random numbers in (0, 1). Genz and Bretz (2002) recommended the use of quasi–random points as good lattice points; see Sloan and Kachoyan (1987) and Hickernell (1998). The GB method is implemented in the R package mvtnorm (Genz and Bretz, 2009).

Nikoloulopoulos (2013) compared the GB method with the simple Monte Carlo approximation proposed by Madsen and Fang (2011), and Nikoloulopoulos (2016) compared the GB method with generalized quantile transform proposed by Kazianka and Pilz (2010) and Kazianka (2013). Both comparisons found the GB method to be more efficient than the competing methods, but these comparisons did not involve the alternative simulated likelihood method described next.

3.2 Geweke–Hajivassiliou–Keane Method

Masarotto and Varin (2012) noticed the similarity between the likelihood function (5) and that of multivariate probit models, and adapted the algorithm proposed by Geweke (1991), Hajivassiliou et al. (1996) and Keane (1994) to compute multivariate normal rectangle probabilities. This is known as the Geweke–Hajivassiliou–Keane simulator, and will be called here the GHK method.

The integral (5) can be approximated by importance sampling, using the importance sampling density with support

$$\left(\Phi^{-1}(F_{\mathbf{s}_1}(y_1 - 1; \boldsymbol{\psi})), \Phi^{-1}(F_{\mathbf{s}_1}(y_1; \boldsymbol{\psi}))\right) \times \cdots \times \left(\Phi^{-1}(F_{\mathbf{s}_n}(y_n - 1; \boldsymbol{\psi})), \Phi^{-1}(F_{\mathbf{s}_n}(y_n; \boldsymbol{\psi}))\right),$$

[1] With the convention that $\sum_{j=1}^{0} v_i = 0$.

given by

$$g_{\eta}(\boldsymbol{z}) = \prod_{i=1}^{n} p_{\eta}(z_i \mid z_{i-1}, \ldots, z_1, y_i),$$

where $p_{\eta}(z_i \mid z_{i-1}, \ldots, z_1, y_i)$ is the conditional density of $Z(\mathbf{s}_i)$ given $Z(\mathbf{s}_{i-1}), \ldots, Z(\mathbf{s}_1)$, and $Y(\mathbf{s}_i) = y_i$, and $Z(\cdot)$ is the Gaussian random field defined after (4). Since $Z(\mathbf{s}_i) \mid Z(\mathbf{s}_{i-1}), \ldots, Z(\mathbf{s}_1) \sim \mathrm{N}(m_i, v_i^2)$, with $m_i = \mathrm{E}_{\vartheta}(Z(\mathbf{s}_i) \mid Z(\mathbf{s}_{i-1}), \ldots, Z(\mathbf{s}_1)) = m\mathbf{i}(Z(\mathbf{s}_{i-1}), \ldots, Z(\mathbf{s}_1); \boldsymbol{\vartheta})$ and $v_i^2 = \mathrm{var}_{\vartheta}(Z(\mathbf{s}_i) \mid Z(\mathbf{s}_{i-1}), \ldots, Z(\mathbf{s}_1)) = v_i^2(\boldsymbol{\vartheta})$, it follows from (4) that $p_{\eta}(z_i \mid Z_{i-1}, \ldots, Z_1, y_i)$ is the density of the $\mathrm{N}(m_i, v_i^2)$ distribution truncated to $(\Phi^{-1}(F_{\mathbf{S}_i}(y_i - 1;\ \psi)), \Phi^{-1}(F_{\mathbf{s}_i}(y_i;\ \psi)))$. Then the GHK method approximates (5) by

$$\begin{aligned} \hat{L}^{\mathrm{GHK}}(\boldsymbol{\eta}; \boldsymbol{y}) &= \frac{1}{M} \sum_{k=1}^{M} \frac{\phi_n(\boldsymbol{Z}^{(k)}; \Psi_{\boldsymbol{\vartheta}})}{g_{\eta}(\boldsymbol{Z}^{(k)})} \\ &= \frac{1}{M} \sum_{k=1}^{M} \left\{ \prod_{i=1}^{n} \frac{p_{\vartheta}(Z_i^{(k)} \mid Z_{i-1}^{(k)}, \ldots, Z_1^{(k)})}{p_{\eta}(Z_i^{(k)} \mid Z_{i-1}^{(k)}, \ldots, Z_1^{(k)}, y_i)} \right\} \\ &= \frac{1}{M} \sum_{k=1}^{M} \left\{ \prod_{i=1}^{n} \left[\Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y_i; \boldsymbol{\psi})) - m_{ki}}{v_{ki}} \right) - \Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y_i - 1; \boldsymbol{\psi})) - m_{ki}}{v_{ki}} \right) \right] \right\}, \end{aligned}$$

where $\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(M)}$ are i.i.d. draws from $g_{\eta}(\mathbf{z})$, with $\boldsymbol{Z}^{(k)} = (Z_1^{(k)}, \ldots, Z_n^{(k)}), m_{ki} = m_i(Z_{i-1}^{(k)}, \ldots, Z_1^{(k)}; \boldsymbol{\vartheta})$ and $v_{ki}^2 = v_i^2(\boldsymbol{\vartheta})$. The simulation of each $\mathbf{Z}^{(k)}$ is done sequentially, $Z_1^{(k)}, Z_2^{(k)}, \ldots, Z_n^{(k)}$, using the standard algorithm for simulating from truncated normal distributions: For $k = 1, \ldots, M$ and $i = 1, \ldots, n$, set

$$Z_i^{(k)} = m_{ki} + v_{ki}\Phi^{-1}\left\{ (1 - U_{ki})\Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y_i - 1; \boldsymbol{\psi}))) - m_{ki}}{v_{ki}} \right) + U_{ki}\Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y_i; \boldsymbol{\psi}))) - m_{ki}}{v_{ki}} \right) \right\},$$

where the U_{ki} are i.i.d. with unif(0, 1) distribution. The GHK method is implemented in the R packages gcmr (Masarotto and Varin, 2017) and gcKrig (Han and De Oliveira, 2018).

Hajivassiliou et al. (1996) compared the GHK method with several other competing methods for the approximation of multivariate normal rectangle probabilities. They found that the GHK method was the most efficient, but this comparison did not involve the GB method nor the method to be described next.

Remark 2. The GHK method requires specifying an ordering of the sampling locations. Although there is a natural ordering in the case of time series data, no natural ordering exists in the case of spatial data, so any choice is somewhat arbitrary. In numerical explorations, we found that when the sampling locations are on a regular grid, the parameter estimates displayed little sensitivity to the chosen ordering. On the other hand, when the sampling locations are on an irregular grid, a common situation with geostatistical data, the parameter estimates displayed a bit more sensitivity for *some* of the chosen orderings, but not all, especially regarding the covariance parameter estimates. Based on our limited experience we conjecture that the sensitivity of the parameter estimates to the chosen ordering of the sampling locations is minor and will decrease as the sample size grows.

4 Data Cloning

Data cloning is a method proposed by Lele et al. (2007), and later studied further by Lele et al. (2010), to compute maximum likelihood estimators (MLE) and their asymptotic standard errors in statistical models with difficulty to compute likelihoods; this is the case for most hierarchical models. The method consists of using a Markov chain Monte Carlo (MCMC) algorithm to carry out an *apparent* Bayesian calculation. Instead of the likelihood based on the observed data, it uses the likelihood that results in the hypothetical scenario that $K > 1$ independent *copies* (clones) of the data had been observed, together with an arbitrary prior on the parameter space. The resulting posterior in the so–called DC–based distribution with the parameter space as its support. The idea is that when K is large, the DC–based distribution becomes highly concentrated at the MLE, regardless of the prior distribution. Then, the MLE can be approximated by the sample mean of draws from the DC–based distribution, and the asymptotic variance of the MLE can be approximated by K times the sample variance of these draws; this will be called the DC method. In spite of its Bayesian motivation, the DC method follows the classical frequentist inferential approach, and uses the MCMC machinery only as a computational device. The DC method was used by Baghishani and Mohammadzadeh (2011) and Torabi (2015) to fit spatial generalized linear mixed models.

4.1 Hierarchical Formulation of Gaussian Copula Models

We show here that the Gaussian copula model (4) can be written as a hierarchical model when the correlation function (2) has a 'nugget effect', i.e., when $\tau^2 > 0$. The Gaussian random field $Z(\cdot)$ admits the representation $Z(\mathbf{s}) \overset{d}{=} \Gamma(\mathbf{s}) + \epsilon(\mathbf{s})$, where $\Gamma(\cdot)$ is a Gaussian random field with mean 0 and covariance function $(1 - \tau^2)\bar{K}_\theta(\mathbf{s}, \mathbf{u})$, $\epsilon(\cdot)$ is Gaussian white noise with $N(0, \tau^2)$ distribution, and $\Gamma(\cdot)$ and $\epsilon(\cdot)$ are independent random fields. Let $\boldsymbol{\Gamma} = (\Gamma(\mathbf{s}_1), \ldots, \Gamma(\mathbf{s}_n))$. Then, the Gaussian copula model for $Y(\cdot)$ is equivalent to the following hierarchical model:

1. For any set of distinct locations $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset D$, the counts $Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_n)$ are conditionally independent given $\boldsymbol{\Gamma}$, and

$$\begin{aligned} Y(\mathbf{s}_i) \mid \boldsymbol{\Gamma} &\overset{d}{=} Y(\mathbf{s}_i) \mid \Gamma(\mathbf{s}_i), \qquad i = 1, \ldots, n \\ &\sim h_i(y \mid \gamma_i; \boldsymbol{\psi}, \tau^2), \end{aligned}$$

where for any $y \in \mathbb{N}_0$

$$h_i(y \mid \gamma_i; \boldsymbol{\psi}, \tau^2) = \Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y; \boldsymbol{\psi})) - \gamma_i}{\tau}\right) - \Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y-1; \boldsymbol{\psi})) - \gamma_i}{\tau}\right); \qquad (9)$$

see Appendix A.

2. $\Gamma(\cdot)$ is Gaussian random field with mean 0 and covariance function $(1-\tau^2)\bar{K}_\theta(\mathbf{s}, \mathbf{u})$, so $\boldsymbol{\Gamma} \sim \mathrm{N}_n(\mathbf{0}, (1-\tau^2)R_\theta)$, with $(R_\theta)_{ij} = \bar{K}_\theta(\mathbf{s}_i, \mathbf{s}_j)$.

4.2 Algorithm

Let $\boldsymbol{y} = (y_1, \ldots, y_n)$ be the data observed at the n sampling locations, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$ an (unobservable) realization of $\boldsymbol{\Gamma}$, which correspond to 'random effects' at the sampling locations, and $\pi(\boldsymbol{\eta}) = \pi(\beta, \sigma^2, \theta, \tau^2)$ the prior distribution of the model parameters, assumed proper. Then, the posterior distribution of $(\boldsymbol{\eta}, \boldsymbol{\gamma})$ based on $\boldsymbol{y}$ is

$$\pi(\boldsymbol{\eta}, \boldsymbol{\gamma} \mid \boldsymbol{y}) \propto \pi(\boldsymbol{\eta})\phi_n(\boldsymbol{\gamma}; (1-\tau^2)R_\theta) \prod_{i=1}^{n} h_i(y_i \mid \gamma_i; \boldsymbol{\psi}, \tau^2),$$

where $h_{\mathrm{i}}(\cdot \mid \gamma_i; \psi, \tau^2)$ is given in (9). Now, let $\boldsymbol{y}_{(K)} = (\boldsymbol{y}, \ldots, \boldsymbol{y})$ be the vector of 'cloned data', formed by concatenating K independent *copies* of the observed counts, and $\boldsymbol{\gamma}_{(K)} = (\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K)$, where $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K \overset{\text{iid}}{\sim} \mathrm{N}_n(\mathbf{0}, (1-\tau^2)R_\theta)$, with $\boldsymbol{\gamma}_k = (\gamma_{k1}, \ldots, \gamma_{kn})$, so γ_{ki} is the random effect at the i^{th} sampling location from the k^{th} replicate of $\boldsymbol{\Gamma}$. Then the 'posterior distribution' of $(\boldsymbol{\eta}, \boldsymbol{\gamma}_{(K)})$ based on $\boldsymbol{y}_{(K)}$ is

$$\begin{aligned} \pi_{(K)}(\boldsymbol{\eta}, \boldsymbol{\gamma}_{(K)} \mid \boldsymbol{y}_{(K)}) &\propto \pi(\boldsymbol{\eta})\pi(\boldsymbol{\gamma}_{(K)} \mid \boldsymbol{\eta})p(\boldsymbol{y}_{(K)} \mid \boldsymbol{\gamma}_{(K)}, \boldsymbol{\eta}) \\ &= \pi(\boldsymbol{\eta}) \prod_{k=1}^{K} \phi_n(\boldsymbol{\gamma}_k; (1-\tau^2)R_\theta) \prod_{k=1}^{K} \prod_{i=1}^{n} h_i(y_i \mid \gamma_{ki}; \boldsymbol{\psi}, \tau^2). \end{aligned} \qquad (10)$$

Let $\hat{\boldsymbol{\eta}}$ be the MLE of $\boldsymbol{\eta}$ based on the observed data $\boldsymbol{y}$. If $\{(\boldsymbol{\eta}^{(l)}, \boldsymbol{\gamma}_{(K)}^{(l)}) : l = 1, \ldots, M\}$ is a large sample from the distribution (10), it follows from results in Lele et al. (2010) that, under suitable conditions, for K large and regardless of the prior $\pi(\boldsymbol{\eta})$

$$\hat{\boldsymbol{\eta}} \approx \frac{1}{M} \sum_{l=1}^{M} \boldsymbol{\eta}^{(l)} \qquad \text{and} \qquad \mathrm{var}(\hat{\boldsymbol{\eta}}) \approx \frac{K}{M-1} \sum_{l=1}^{M} (\boldsymbol{\eta}^{(l)} - \hat{\boldsymbol{\eta}})(\boldsymbol{\eta}^{(l)} - \hat{\boldsymbol{\eta}})^\intercal,$$

where these approximations improve as $K \to \infty$; see also Walker (1969) and Baghishani and Mohammadzadeh (2011) for similar results. Unlike simulated likelihood methods, the DC method approximates $\hat{\boldsymbol{\eta}}$ and its (asymptotic) variance directly, without approximating the likelihood function.

The number of clones K is chosen to strike a balance between computational accuracy and effort. Lele et al. (2010) proposed a simple rule to diagnose whether the DC–based distribution is close to 'degenerate', based on the fact that as $K \to \infty$ the largest eigenvalue of the variance matrix of the DC–based distribution converges to zero at the same rate as 1/

K. They proposed plotting λ_K/λ_1 versus K, $K = 1, 2, \ldots$, where λ_K is the largest eigenvalue of the variance matrix of the DC–based distribution based on K clones, and comparing this plot with the plot of $1/K$. The number of clones is chosen as the smallest K for which λ_K/λ_1 falls below a small positive threshold, say 0.1.

We use priors of the form $\pi(\boldsymbol\beta, \sigma^2, \boldsymbol\theta, \tau^2) = \pi(\boldsymbol\beta)\pi(\sigma^2)\pi(\boldsymbol\theta)\pi(\tau^2)$, where the marginal priors are chosen to be informative, as such priors require fewer clones than non–informative priors to achieve a desired accuracy; the MLE obtained by the DC method is insensitive to the chosen prior. Any Monte Carlo algorithm that produces samples from (10) would work. We found, after some experimentation, that the following ‘Metropolis–within–Gibbs’ sampler algorithm works well. The state variables $(\boldsymbol\beta, \sigma^2, \boldsymbol\theta, \tau^2, \gamma_{(K)})$ are updated by separately updating two blocks, $(\boldsymbol\beta, \sigma^2)$ and $(\boldsymbol\theta, \tau^2, \gamma_{(K)})$, where each block is updated using a Metropolis–Hastings algorithm. The details of the algorithm are given in Appendix B.

The requirement that the covariance function of the latent process $Z(\cdot)$ includes a nugget effect is a limitation of the above MCMC algorithm for implementing the DC method, arguably a moderate one since geostatistical data often contain measurement error. Although a model with a fixed small nugget effect could be used to approximately fit data with no measurement error, this is likely to produce computational instabilities in the evaluation of $h_i(\cdot \mid \gamma_i; \psi, \tau^2)$.

5 Initial Values

All of the previously described methods require good initial values for the parameters for an efficient implementation. Good initial values would produce fast convergence of the iterative algorithm used for likelihood optimization in the GB and GHK methods, and good initial values allow the use of informative priors in the DC method.

We use a simple two–stage estimation method to obtain initial values, which shares some similarities with one of the methods proposed by Zhao and Joe (2005). The first stage consists of estimating the marginal parameters by maximizing the pseudo–likelihood obtained under the working assumption of independence, resulting in

$$\widehat{\boldsymbol\psi} := \arg\max_{\boldsymbol\psi} \prod_{i=1}^{n} f_{\mathbf{s}_i}(y_i; \boldsymbol\psi).$$

Next, note from (4) that for any $\mathbf{s} \in D$ and $k \in \mathbb{N}_0$

$$Z(\mathbf{s}) \mid Y(\mathbf{s}) = k \sim \mathrm{N}(0,1) \text{ truncated to } (\Phi^{-1}(F_{\mathbf{s}}(k-1;\boldsymbol\psi)), \Phi^{-1}(F_{\mathbf{s}}(k;\boldsymbol\psi))),$$

so a kind of ‘residuals’ can be defined as

$$\widehat{Z}_i := \mathrm{E}_{\widehat{\boldsymbol\psi}}\{Z(\mathbf{s}_i) \mid Y(\mathbf{s}_i) = y_i\} = \frac{\phi(\Phi^{-1}(F_{\mathbf{s}_i}(y_i-1;\widehat{\boldsymbol\psi}))) - \phi(\Phi^{-1}(F_{\mathbf{s}_i}(y_i;\widehat{\boldsymbol\psi})))}{F_{\mathbf{s}_i}(y_i;\widehat{\boldsymbol\psi}) - F_{\mathbf{s}_i}(y_i-1;\widehat{\boldsymbol\psi})}, \qquad i = 1, \ldots, n,$$

where $\varphi(\cdot)$ is the pdf of the standard normal distribution. The second stage consists of estimating the covariance parameters by maximizing the pseudo–likelihood obtained under the working assumption that the joint distribution of $\widehat{\boldsymbol{Z}} = (\widehat{Z}_1, \ldots, \widehat{Z}_n)^\top$ is Gaussian with mean vector $\mathbf{0}$ and covariance matrix Ψ_{ϑ}, resulting in

$$\widehat{\boldsymbol{\vartheta}} := \arg\max_{\boldsymbol{\vartheta}} |\Psi_{\boldsymbol{\vartheta}}|^{-1/2} \exp\{-\frac{1}{2}\widehat{Z}^\top \Psi_{\boldsymbol{\vartheta}}^{-1} \widehat{Z}\}.$$

Remark 3. The GB and GHK methods provide approximations to likelihood functions at the MLEs (or any other parameter values), so they can be used for likelihood–based inferential tasks such as model selection, testing of nested hypotheses and computation of likelihood–based confidence intervals. Although the DC method does not approximate likelihood functions directly, likelihoods and likelihood ratios can be approximated using Monte Carlo techniques described in Thompson (1994), as long as the models under study are hierarchical. Ponciano et al. (2009) described how to perform the aforementioned likelihood–based inferential tasks using MLEs obtained from the DC method and a sample from the conditional distribution of the random effects given the data and parameter estimates. An application of some of these inferential tasks will be illustrated in Section 7.

6 Simulation Study

We conduct a simulation study to compare the three methods described above, GB, GHK and DC, to approximate the MLEs of the model parameters, and investigate some of their statistical properties. The properties to be investigated are absolute relative bias and mean absolute error, as well as coverage and length of different types of confidence intervals. In addition, we also compare the required computational effort to implement these methods. The results for the GB and GHK methods were obtained using, respectively, the R packages mvtnorm (Genz and Bretz, 2009) and gcKrig (Han and De Oliveira, 2018), while the DC method was coded and implemented by us in R and C++.

6.1 Design

Below we describe the factors of the simulation design, where some are fixed and others vary, as well as the MCMC settings required for the DC method.

- The study region is $D = [0, 1] \times [0, 1]$. The sampling locations form an 11×11 regular lattice within D, so $n = 121$ and the minimum distance between sampling locations is 0.1.
- As marginals we use the family of negative binomial distributions parameterized in terms of their mean and a 'size' parameter (the so–called NB2 distribution; Cameron and Trivedi (2013)), where the pmf of $Y(\mathbf{s})$ is given by

$$f_{\mathbf{s}}^{\mathrm{NB}}(y; \boldsymbol{\psi}) = \frac{\Gamma(y + 1/\sigma^2)}{\Gamma(1/\sigma^2)y!}\left(\frac{1}{1 + \sigma^2\mu(\mathbf{s})}\right)^{1/\sigma^2}\left(1 - \frac{1}{1 + \sigma^2\mu(\mathbf{s})}\right)^{y}, \quad y = 0, 1, 2, \ldots \qquad (11)$$

With this parameterization it holds that E($Y(\mathbf{s})$) = $\mu(\mathbf{s})$ and var($Y(\mathbf{s})$) = $\mu(\mathbf{s})(1 + \sigma^2\mu(\mathbf{s}))$.

- The mean function $\mu(\mathbf{s})$ is exp(0.5 + 0.5x + y), so $(\beta_1, \beta_2, \beta_3) = (0.5, 0.5, 1)$. The marginal priors for $\boldsymbol{\beta}$ used in the DC method is $N_3((0.5, 0.5, 1)^\top, 30I_3)$.
- The dispersion parameter σ^2 is either 0.2 (small overdispersion) or 2 (large overdispersion). The marginal priors for σ^2 used in the DC method are IG(0.2,0.4) (inverse Gaussian) and IG(2, 4), respectively.
- The continuous correlation function $\bar{K}_\theta(\mathbf{s}, \mathbf{u})$ in (2) is either exp(−d/0.1) or exp(−d/0.3), with $d = \|\mathbf{s} - \mathbf{u}\|$ being Euclidean distance, so $\theta = 0.1$ (weak dependence) or $\theta = 0.3$ (strong dependence). The marginal priors for θ used in the DC method are LN(−2, 1) (lognormal) and LN(−1, 0.5), respectively.
- The nugget effect τ^2 in the correlation function (2) is 0.25, and its marginal prior used in the DC method is unif(0, 1).
- The number of clones for the DC method is $K = 40$. This choice was made based on plots of λ_K/λ_1 versus K obtained from a few pilot runs; see Section 4.2 and Figure 1.
- For the MCMC algorithm in the DC method, $\boldsymbol{\eta}^{(0)}$ were set at the initial values described in Section 5. The tuning constants $c_1, c_2, c_3, c_4 > 0$ were updated every 1000 iterations with memory parameter equal to 1000, while the tuning constant c_5 was fixed at 0.03; see Appendix B for details. The length of each MCMC run is 15000, with the first 5000 iterations as 'burn–in'. These choices ensure the automaticity and robustness of the simulations.

For each of the four simulation scenarios (two values of σ^2 and two values of θ) we simulated 1000 datasets, each of size $n = 121$, and computed the approximate MLE using the three methods. The approximate MLEs will be compared in terms of absolute relative bias (ARB) and mean absolute error (MAE). These are defined, for each simulation scenario, method and parameter 'a', as

$$\text{ARB} = \left| \frac{\text{MLE} - a_{\text{true}}}{a_{\text{true}}} \right| \quad \text{and} \quad \text{MAE} = \frac{1}{1000} \sum_{i=1}^{1000} |\, \hat{a}_i - a_{\text{true}} \,| ,$$

where a_{ture} is the true value of a, $\hat{a}_i$ is the estimate of a_{ture} based on the i^{th} simulated dataset, 1000 and MLE = $\frac{1}{1000}\sum_{i=1}^{1000}\hat{a}_i$. In addition, for each simulation scenario we also compare the coverage and length of three types of (approximate) 95% confidence intervals (see below), defined as

$$\text{coverage} = \frac{1}{1000} \sum_{i=1}^{1000} \mathbf{1}\{\text{LB}_i < a_{\text{true}} < \text{UB}_i\}, \ \text{length} = \text{median}\{\text{UB}_i - \text{LB}_i : i = 1, \ldots, 1000\},$$

where LB_i, UB_i are the endpoints of a confidence interval based on the i^{th} simulated dataset. The 95% confidence intervals to be considered are:

- Wald: The Wald–type confidence interval has endpoints $(\mathrm{LB}_i, \mathrm{UB}_i) = \hat{a}_i \mp 1.96(\hat{\mathrm{var}}(\hat{a}_i))^{1/2}$, where $\hat{\mathrm{var}}(\hat{a}_i)$ is an estimate of the (asymptotic) variance of $\hat{a}_i$ based on the ith simulated dataset. For the GHK method the variance estimates are obtained from the diagonal elements of the (estimated) inverse Hessian matrix obtained at the end of the optimization process.
- Wald–Tran: This is a Wald–type confidence interval obtained using the reparametrization of the likelihood in terms of the original regression parameters $(\beta_1, \beta_2, \beta_3)$ and the transformed parameters $(\xi_2, \xi_3, \xi_4) = (\log(\sigma^2), \log(\theta), \mathrm{logit}(\tau^2))$. The confidence intervals for the original parameters are obtained by back–transforming the endpoints of the Wald–type confidence intervals for the transformed parameters.
- PL: This is the confidence interval obtained by evaluating the profile likelihood and inverting a likelihood ratio test (Meeker and Escobar, 1995). LB_i and UB_i can be computed either by evaluating the profile likelihood on a selected grid of parameter values or by evaluating the profile likelihood over parameter values selected by an iterative algorithm; gcmr uses the former while gcKrig uses the latter.

6.2 Results

Table 1 displays the ARB and MAE results. Overall, the three methods to approximate the maximum likelihood estimators are comparable, in the sense that they produce similar ARB and MAE. In regard to the regression parameters, all methods produce MLEs for β_2 and β_3 that are close to being unbiased, but the MLEs of β_1 are (downward) biased when the dependence is strong. All the methods produce MLEs for σ^2, θ and τ^2 that are (downward) biased, with the ARB tending to increase when either the overdispersion or strength of dependence increase. Also, except for τ^2, the MAE of the MLEs of all parameters increase with overdispersion and strength of dependence.

Next, for each simulation scenario and parameter we computed the three types of (approximate) 95% confidence intervals described above, and compare them in terms of their coverage and length. This is done only for the GHK method since the other methods produce very similar results.

Table 2 displays the confidence intervals results. For the regression parameters, all the confidence intervals display undercoverage, and this undercoverage becomes more substantial as the strength of dependence increases. Among all the confidence intervals the profile likelihood confidence intervals have coverage that are closest to nominal, while that of the other two are similar. The profile likelihood confidence intervals are in general wider, which partly explain their better coverage.

For the overdispersion parameter, all confidence intervals tend to display undercoverage, except when both dependence and overdispersion are small. The undercoverage of the

Wald–Tran and profile likelihood confidence intervals are smaller than that of the Wald confidence interval, an the length of the all confidence intervals are similar, with Wald–Tran being slightly larger sometimes. For the range parameter, the Wald confidence interval displays overcoverage when the strength of dependence is small, and undercoverage when the strength of dependence is large. The Wald–Tran confidence interval always displays undercoverage. The profile likelihood confidence interval has coverage that is closest to nominal, and is wider than the other two confidence intervals. Finally for the nugget parameter, all three confidence intervals have similar coverages that are close to nominal, and the Wald confidence interval is wider than the profile likelihood confidence interval. Overall, profile likelihood confidence intervals seem to perform better than the other two, in agreement with what has been observed in other models and settings; see Meeker and Escobar (1995).

Although the above properties of point and interval estimators are somewhat unsatisfactory, they are in agreement with properties of maximum likelihood parameter inferences in Gaussian random fields; see Irvine et al. (2007) and the references therein. We expect the above properties of point and interval estimators to improve as the sample size grows in an increasing domain asymptotic regime (but not necessarily so in an infill asymptotic regime). To investigate this conjecture, we ran a second simulation study with the same design as before, except for the factor in the first bullet. Now we consider the larger study region $D = [0, 2] \times [0, 2]$, where the sampling locations form a 21 × 21 regular lattice so $n = 441$ and the minimum distance between sampling locations is still 0.1.

Table 3 displays the combined results for point and interval estimators using the GHK method (results for Wald confidence intervals are not reported since they are similar to or inferior than the results for Wald–Tran confidence intervals.) The MLEs for this larger sample size are, for most parameters and scenarios, approximately unbiased and the MAE are substantially reduced. The coverage probabilities of both confidence intervals are, for most parameters and scenarios, closer to nominal than those for the smaller sample size. The exception occurs for the Wald–Tran confidence interval for the range parameter that displays severe undercoverage for some scenarios. For most parameters and scenarios, the profile likelihood confidence intervals have coverages that are similar or better than those of the Wald–Tran confidence intervals, and theirs lengths are similar. Again, the profile likelihood confidence intervals are preferable. We end by pointing out that we did not include the GB and DC methods in this larger simulation study because their required computational effort is substantially much larger than that of the GHK method; see the next subsection.

6.3 Computational Time

We now compare the three methods to approximate the MLE in terms of computational effort. The conclusions derived from such comparisons are arguably tentative, since computational times depend heavily on many implementation factors that are bound to change (evolve), such as programming environment and language, the method of optimization and even starting points in the optimization. Nevertheless, the following comparison provides information about the relative computational efforts required for the three methods under their current implementations, and how these efforts increase with sample size.

For the GB method the maximum number of function evaluations (maxpts) in mvtnorm was set to 25000, while the choices of M and P are selected (optimized) automatically by the package. For the GHK method the Monte Carlo size was set at $M = 1000$. For both simulated likelihood methods the R function optim was used with box constrained quasi-Newton method for optimization method = "L-BFGS-B" using same initial values. For the DC method the heavy-duty number crunching, like generation of proposal distributions, was coded in C++ with interface to R using the packages Rcpp and RcppArmadillo (Eddelbuettel and Sanderson, 2014), and the MCMC iterations were coded in R. We controlled other factors so the computational environment under different methods is the same. All computations were carried out on a Macbook Pro computer with a 2.8 GHz Intel Core i7 processor, 16 GB RAM and 6 MB L3 Cache using 4 cores working in parallel. To speed up the computations we used the R package snowfall (Knaus, 2015) to parallelize the computations in the three methods.

Table 4 displays the times (in seconds) that took for each of the three methods to compute the average of the MLE from 100 simulations with different sample sizes. Clearly, the GHK simulator is substantially faster than its competitors, more than ten times faster on average. It is faster than the GB method due to the generation of quasi–random points required for the latter, at least for the tested sample sizes (the maximum allowed dimension in the current version of mvtnorm is $n = 1000$). On the other hand, the DC method is the most computationally intensive due to its use of MCMC and clones of data. Given these findings and the results in Table 1, we conclude that the GHK method provides the best balance between statistical and computational efficiency.

7 Example: Lansing Woods Dataset

In this section we illustrate the application of the proposed methods for the analysis of a dataset previously analyzed by Kazianka (2013) using other methods. For comparison we follow and extend his analyses. The original dataset consists of the locations of 2251 trees and their botanical classification as hickories, maples, red oaks, white oaks, black oaks or miscellaneous trees, within a plot of 924 × 924 feet in Lansing Woods, Michigan, USA; it is available in the R package spatstat. The question of interest is to determine whether the presence of hickory trees tend to discourage the presence of maple trees nearby. These data were summarized by rescaling the original plot into the unit square and dividing the latter into 256 square quadrats of equal area. Let $Y(\mathbf{s}_i)$ denote the number of maple trees and $f_2(\mathbf{s}_i)$ the number of hickory trees in the quadrat with center $\mathbf{s}_i$, $i = 1,\ldots, 256$. Figure 2 displays the spatial distribution of these variables.

As in Kazianka (2013), we first fit the Gaussian copula model with Poi($\exp(\beta_1 + \beta_2 f_2(\mathbf{s}))$) marginals and correlation function $K_\vartheta(\mathbf{s}, \mathbf{u}) = (1 - \tau^2)\exp(-d/\theta) + \tau^2 \mathbf{1}\{\mathbf{s} = \mathbf{u}\}$. This model is fit using two methods, GHK and DC, where for the latter we used 40 clones and the priors $\boldsymbol{\beta} \sim \mathrm{N}_2(\widehat{\boldsymbol{\beta}}_{\mathrm{INI}}, 30 I_2)$, $\theta \sim \mathrm{LN}(-2, 1)$ and $\tau^2 \sim \mathrm{unif}(0,1)$, with $\widehat{\boldsymbol{\beta}}_{\mathrm{INI}}$ the initial value of $\boldsymbol{\beta}$ whose computation is described in Section 5. The GHK implementation is carried out using the R package gcKrig. Table 5 displays the MLE and their standard errors (in parentheses) for each method, as well as the estimates obtained by Kazianka (2013) using two other estimation methods (generalized quantile transform (GQT) and composite likelihood (CL)). The estimates from both methods are quite similar for all parameters, and the standard errors display small discrepancies. For both methods the Wald–type 99% confidence interval for β_2 is made out entirely of negative numbers, suggesting the counts of hickory and maple trees in the quadrats are negatively associated. These findings agree with those in Kazianka (2013).

To investigate the adequacy of the Poisson family of marginal distributions, we also fit Gaussian copula models with negative binomial distributions, as in (11), and zero–inflated Poisson (ZIP) distributions as marginals, the latter having pmf given by

$$f_{\mathbf{s}}^{\text{ZIP}}(y;\boldsymbol{\psi}) = \begin{cases} \frac{\sigma^2}{\sigma^2+1} + \frac{1}{\sigma^2+1}\exp(-(1+\sigma^2)\mu(\mathbf{s})) & \text{if } y = 0 \\ \frac{1}{\sigma^2+1}\frac{\exp(-(1+\sigma^2)\mu(\mathbf{s}))((1+\sigma^2)\mu(\mathbf{s}))^y}{y!} & \text{if } y = 1, 2, \ldots \end{cases};$$

see Han and De Oliveira (2016). Both families are parameterized in terms of the mean function $\mu(\mathbf{s}) = \exp(\beta_1 + \beta_2 f_2(\mathbf{s}))$ and a 'size' parameter $\sigma^2 \geq 0$ that controls overdispersion, and both include the aforementioned family of Poisson distributions (that results when $\sigma^2 = 0$). We continue using the aforementioned exponential correlation function with nugget $\tau^2 \in [0, 1]$.

Table 6 displays the MLE and asymptotic standard errors (in parenthesis) using the GHK and DC methods under both families of marginals. To fit the models with negative binomial and zero-inflated Poisson family of marginals using the DC method we used the same priors as in the fit with Poisson marginals, plus $\sigma^2 \sim \text{IG}(1,1)$. The parameter estimates under Poisson and zero–inflated Poisson marginals are similar, while some differences are apparent under the negative binomial marginals. Under these families of marginals the Wald–type 99% confidence interval for β_2 are also made out entirely of negative numbers, supporting again the conclusion that there exist negative association between the number of hickory and maple trees in the quadrats. In addition, the estimates of σ^2 are positive and away from zero for both methods and both families of marginals, supporting the presence of overdispersion.

As indicated in Remark 3, the GHK method approximates the likelihood function evaluated at the MLE, so information criteria can be used for selecting among candidate models. Table 7 reports the log–likelihood, AIC$_c$ and BIC values for the fitted Gaussian copula models with families of Poisson, ZIP and NB2 marginals. The model with NB2 marginal is the one selected for both the AIC$_c$ and BIC criteria. Based on the DC method, Ponciano et al. (2009) described a Monte Carlo method to approximate likelihood ratios which can be used to compute the difference in information criteria between two competing models, say BIC$^{\text{model}_1}$ – BIC$^{\text{model}_2}$, which reaches the same conclusion. Neither the models with Poisson or ZIP marginals are competitive, so the model with negative binomial marginals seems the more appropriate for this dataset.

To assess the adequacy of the fitted Gaussian copula model with negative binomial marginals selected above we use the kind of residuals advocated by Masarotto and Varin (2012). Specifically, we use the randomized quantile residuals defined as

$$r_i = \Phi^{-1}\{U_i F_{\mathbf{s}_i}(y_i \mid y_{i-1}, \ldots, y_1; \widehat{\boldsymbol{\eta}}) + (1 - U_i) F_{\mathbf{s}_i}(y_i - 1 \mid y_{i-1}, \ldots, y_1; \widehat{\boldsymbol{\eta}})\},$$

for $i = 1, \ldots, n$, where $\widehat{\boldsymbol{\eta}} = (\hat{\beta}_1, \hat{\beta}_2, \hat{\sigma}^2, \hat{\theta}, \hat{\tau}^2)$, $F_{\mathbf{s}_i}(\cdot \mid y_{i-1}, \ldots, y_1; \boldsymbol{\eta})$ is the conditional cdf of $Y(\mathbf{s}_i)$ given $Y(\mathbf{s}_{i-1}), \ldots, Y(\mathbf{s}_1)$ and $U_1, \ldots, U_n$ is an independent random sample from the unif(0,1) distribution; these are computed using the R package gcmr. If the model is correctly specified, then $r_1, \ldots, r_n$ behave (approximately) as a random sample from the standard normal distribution. Figure 3 displays different graphical summaries of these residuals. All of them support the normality assumption, and consequently the adequacy of the fitted model.

Finally, likelihood–based inferences are possible using the GHK method. Figure 4 displays the profile log–likelihoods of β_2 and σ^2 under the NB2 model, from which follow that (approximate) 99% confidence intervals for these parameters are (−0.210, −0.019) and (0.534, 5.012), respectively; the corresponding Wald–type 99% confidence intervals are (−0.206, −0.018) and (0.031, 2.524). These support even more strongly the above conclusions regarding the presence of a negative association between the number of hickory and maple trees in the quadrats, and the existence of overdispersion in the marginal distributions of the counts.

A script with code to replicate the data analysis of this section can be obtained from https://github.com/hanzifei/Code_GC_MLE.

8 Conclusions

In this work we considered the computationally challenging task of (numerically) computing MLE of parameters in Gaussian copula models for geostatistical count data. We reviewed two previously proposed Monte Carlo methods to approximate the likelihood function, the GB and GHK methods, and investigated the implementation for this model of the DC method. We also proposed a simple and fast way to obtain initial values that are needed for all the methods.

Based on our simulation study, the statistical properties of the MLE approximated by the three methods are similar. Of the three types of confidence intervals we explored, the profile likelihood confidence intervals displayed the best coverage, at the price of being on average wider, while the Wald–type confidence intervals displayed the worst coverage. Overall, we recommend using the former. The estimation bias of MLE and undercoverage of confidence intervals can be substantial when the sample size is small, but these become better as the sample size grows following an increasing domain asymptotic regime.

In terms of computational effort, the GHK method is the least computationally intensive of the three methods, while the GB and DC methods are much more computationally intensive, about ten times more on average. Because of all of the above, the GHK method provides the best balance between statistical efficiency and computational efficiency, so it is the method we recommend. Nevertheless, it is worth noting the ‘hybrid’ method proposed by Baghishani et al. (2012) that blends the DC method and integrated nested Laplace approximations. That article showed that the computational effort of this hybrid method is substantially smaller than that of the DC methods when fitting some simple hierarchical models, so investigating the properties of this hybrid method for fitting Gaussian copula models to geostatistical count data seems a worthwhile research problem.

Finally, note that spatial prediction/interpolation is often the end goal of many geostatistical analyses. For parameter estimates $\hat{\boldsymbol{\eta}}$ and prediction location s0, the (plug–in) predictive distribution $\mathrm{P}_{\hat{\eta}}(Y(\mathbf{s}_0) = y_0 \mid \boldsymbol{Y} = \boldsymbol{y})$ is the ratio of two multivariate integrals like that in (5), so these can also be approximated using the GHK method. Details on this prediction method are given in Han and De Oliveira (2018).

Acknowledgements.

We warmly thank two anonymous referees for helpful comments and suggestions that lead to an improved article. This work was partially supported by the U.S. National Science Foundation Grant DMS–1208896, and received computational support from Computational System Biology Core, funded by the National Institute on Minority Health and Health Disparities (G12MD007591) from the National Institutes of Health.

Appendix A

Proof of Identity (7)

For the inverse of the map (6) we have

$$\frac{d}{du_i} z_i(u_i) = \frac{-f_{\mathbf{s}_i}(y_i; \boldsymbol{\psi})}{\phi(\Phi^{-1}(F_{\mathbf{s}_i}(y_i; \boldsymbol{\psi}) - u_i f_{\mathbf{s}_i}(y_i; \boldsymbol{\psi}))} = \frac{-f_{\mathbf{s}_i}(y_i; \boldsymbol{\psi})}{\phi(z_i(u_i))}, \qquad i = 1, \ldots, n\,.$$

Since z_i is a function of u_i alone, the Jacobian matrix $J = \partial(z_1(u_1), \ldots, z_n(u_n))/\partial(u_1, \ldots, u_n)$ of the inverse mapping is diagonal, so

$$\mid \det(J) \mid \; = \prod_{i=1}^{n} \frac{f_{\mathbf{s}_i}(y_i; \boldsymbol{\psi})}{\phi(z_i(u_i))}\,.$$

Since (6) maps $(\Phi^{-1}(F_{\mathbf{s}_1}(y_1 – 1;\ \psi)), \Phi^{-1}(F_{\mathbf{s}_1}(y_1;\ \psi))) \times \cdots \times (\Phi^{-1}(F_{\mathbf{s}_\mathrm{n}}(y_n – 1;\ \psi)), \Phi^{-1}(F_{\mathbf{s}n}(y_n;\ \psi)))$ into $(0, 1)^n$, by the change of variable formula for multivariate integrals, (5) becomes

$$\begin{aligned} L(\boldsymbol{\eta}; \boldsymbol{y}) &= \int_0^1 \cdots \int_0^1 \phi_n(z_1(u_1), \ldots, z_n(u_n); \Psi_{\boldsymbol{\vartheta}}) \mid \det(J) \mid \ du_1 \ldots du_n \\ &= \prod_{i=1}^{n} f_{\mathbf{s}_i}(y_i; \boldsymbol{\psi}) \cdot \int_0^1 \cdots \int_0^1 \frac{\phi_n(z_1(u_1), \ldots, z_n(u_n); \Psi_{\boldsymbol{\vartheta}})}{\prod_{i=1}^{n} \phi(z_i(u_i))} du_1 \ldots du_n \\ &= \prod_{i=1}^{n} f_{\mathbf{s}_i}(y_i; \boldsymbol{\psi}) \cdot \int_0^1 \cdots \int_0^1 \mid \Psi_{\boldsymbol{\vartheta}} \mid^{-1/2} \exp\left(-\frac{1}{2} \boldsymbol{z}^\intercal(\mathbf{u})(\Psi_{\boldsymbol{\vartheta}}^{-1} - \boldsymbol{I}_n)\boldsymbol{z}(\mathbf{u})\right) d\mathbf{u}\,. \end{aligned}$$

Equivalence Between (7) and Madsen and Fang Likelihood Representation

Madsen and Fang (2011) showed that the likelihood in (5) can be written as

$$L(\boldsymbol{\eta}; \boldsymbol{y}) = \mathrm{E}_{\boldsymbol{U}}\left\{ \mid \Psi_{\vartheta} \mid^{-1/2}\exp\left(-\frac{1}{2}\boldsymbol{z}^{*\top}(\boldsymbol{U})(\Psi_{\vartheta}^{-1} - \boldsymbol{I}_n)\boldsymbol{z}^*(\boldsymbol{U})\right)\cdot \prod_{i=1}^{n} f^*_{\mathbf{s}_i}(y_i - U_i; \boldsymbol{\psi})\right\}, \qquad (12)$$

where

$$\boldsymbol{z}^*(\boldsymbol{U}) = \left(\Phi^{-1}(F^*_{\mathbf{s}_1}(y_1 - U_1; \boldsymbol{\psi})), \ldots, \Phi^{-1}(F^*_{\mathbf{s}_n}(y_n - U_n; \boldsymbol{\psi}))\right)^{\top},$$

$\mathbf{U} = (U_1,\ldots,U_n) \sim \mathrm{unif}((0,1)^n)$, and $f^*_{\mathbf{s}_i}(\cdot\ ; \boldsymbol{\psi})$ and $F^*_{\mathbf{s}_i}(\cdot\ ; \boldsymbol{\psi})$ are, respectively, the pdf and cdf of the 'continuously extended' random variable $Y^*_i = Y(\mathbf{s}_i) - U_i$. Now, it can be shown that for every $\mathbf{s} \in D$ and $y \in \mathbb{R}$, $F^*_{\mathbf{s}}(y; \boldsymbol{\psi}) = F_{\mathbf{s}}(\lfloor y \rfloor; \boldsymbol{\psi}) + (y - \lfloor y \rfloor) f_{\mathbf{s}}(\lfloor y + 1 \rfloor; \boldsymbol{\psi})$ and $f^*_{\mathbf{s}}(y; \boldsymbol{\psi}) = f_{\mathbf{s}}(\lfloor y + 1 \rfloor; \boldsymbol{\psi})$, where $[y]$ is the largest integer not greater than y. Hence for any $y \in \mathbb{N}_0$ and $u \in (0, 1)$, we have $f^*_{\mathbf{s}}(y - u; \boldsymbol{\psi}) = f_{\mathbf{s}}(y; \boldsymbol{\psi})$ and $F^*_{\mathbf{s}}(y - u; \boldsymbol{\psi}) = F_{\mathbf{s}}(y - 1; \boldsymbol{\psi}) + (1 - u) f_{\mathbf{s}}(y; \boldsymbol{\psi}) = F_{\mathbf{s}}(y; \boldsymbol{\psi}) - u f_{\mathbf{s}}(y; \boldsymbol{\psi})$. Substituting these into(12) shows that the right–hand sides of (12) and (7) are the same.

Proof of Identity (9)

As indicated in Section 2.2, the events $\{Y(\mathbf{s}) = y\}$ and $\{F_{\mathbf{s}}(y - 1;\ \psi) < \Phi(Z(\mathbf{s})) \le F_{\mathbf{s}}(y;\ \psi)\}$ are equivalent, so for $y \in \mathbb{N}_0$ and $i = 1,\ldots,n$ we have

$$\begin{aligned}
& \mathrm{P}(Y(\mathbf{s}_i) = y \mid \Gamma(\mathbf{s}_i) = \gamma_i; \boldsymbol{\psi}, \tau^2) \\
= \quad & \mathrm{P}(\Phi^{-1}(F_{\mathbf{s}_i}(y - 1; \boldsymbol{\psi})) < Z(\mathbf{s}_i) \le \Phi^{-1}(F_{\mathbf{s}_i}(y; \boldsymbol{\psi})) \mid \Gamma(\mathbf{s}_i) = \gamma_i; \boldsymbol{\psi}, \tau^2) \\
= \quad & \mathrm{P}\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y - 1; \boldsymbol{\psi})) - \gamma_i}{\tau} < \frac{\epsilon(\mathbf{s}_i)}{\tau} \le \frac{\Phi^{-1}(F_{\mathbf{s}_i}(y; \boldsymbol{\psi})) - \gamma_i}{\tau} \,\middle|\, \Gamma(\mathbf{s}_i) = \gamma_i; \boldsymbol{\psi}, \tau^2\right) \\
= \quad & \Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y; \boldsymbol{\psi})) - \gamma_i}{\tau}\right) - \Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y - 1; \boldsymbol{\psi})) - \gamma_i}{\tau}\right).
\end{aligned}$$

Appendix B

MCMC Algorithm for the DC Method

In the Gaussian copula model, $\boldsymbol{\beta} \in \mathbb{R}^p$, θ, $\sigma^2 > 0$ and $\tau^2 \in [0, 1]$. To ease updating the parameters using a random–walk Metropolis–Hastings algorithm with Gaussian proposals, we transform each parameter to the real line. Specifically, let $\boldsymbol{\xi}_1 = \boldsymbol{\beta}$, $\xi_2 = \log(\sigma^2)$, $\xi_3 = \log(\theta)$ and $\xi_4 = \mathrm{logit}(\tau^2)$. As customary in the statistical literature, $q_a(\cdot \mid a^{\mathrm{old}})$ would denote the proposal distribution for the state component 'a' in the current iteration, given its value in the previous iteration is a^{old}. The algorithm proceeds as follows:

Step 0: Choose the initial values for the parameters $\boldsymbol{\eta}^{(0)}$; simulate $\boldsymbol{\gamma}_1^{(0)}, \ldots, \boldsymbol{\gamma}_K^{(0)} \overset{\text{iid}}{\sim} \mathrm{N}_n(\mathbf{0}, R_{\theta(0)})$ and set the initial values for the replicated latent process as $\boldsymbol{\gamma}_{(K)}^{(0)} = (\boldsymbol{\gamma}_1^{(0)}, \ldots, \boldsymbol{\gamma}_K^{(0)})$; set the tuning constants c_1, c_2, c_3, c_4, $c_5 > 0$; set $l = 0$.

Step 1: Simulate independently $\boldsymbol{\beta}^* \sim \mathrm{N}_p(\boldsymbol{\beta}^{(l)}, c_1 I_p)$, $\xi_2^* \sim \mathrm{N}(\xi_2^{(l)}, c_2)$ and $U_1 \sim \text{unif}(0, 1)$.

Step 2: Compute the acceptance probability $\alpha_1 = \min\{1, t_1\}$, where

$$\begin{aligned} t_1 &= \frac{\pi_{(K)}(\boldsymbol{\beta}^*, \sigma^{2*} \mid \theta^{(l)}, \tau^{2(l)}, \boldsymbol{\gamma}_{(K)}^{(l)}, \mathbf{y}_{(K)}) q_\beta(\boldsymbol{\beta}^{(l)} \mid \boldsymbol{\beta}^*) q_{\sigma^2}(\sigma^{2(l)} \mid \sigma^{2*})}{\pi_{(K)}(\boldsymbol{\beta}^{(l)}, \sigma^{2(l)} \mid \theta^{(l)}, \tau^{2(l)}, \boldsymbol{\gamma}_{(K)}^{(l)}, \mathbf{y}_{(K)}) q_\beta(\boldsymbol{\beta}^* \mid \boldsymbol{\beta}^{(l)}) q_{\sigma^2}(\sigma^{2*} \mid \sigma^{2(l)})} \\ &= \frac{\pi(\boldsymbol{\beta}^*)\pi(\sigma^{2*})\prod_{k=1}^K \prod_{i=1}^n h_i(y_i \mid \gamma_{ki}^{(l)}; \boldsymbol{\beta}^*, \sigma^{2*}, \tau^{2(l)})\sigma^{2*}}{\pi(\boldsymbol{\beta}^{(l)})\pi(\sigma^{2(l)})\prod_{k=1}^K \prod_{i=1}^n h_i(y_i \mid \gamma_{ki}^{(l)}; \boldsymbol{\beta}^{(l)}, \sigma^{2(l)}, \tau^{2(l)})\sigma^{2(l)}}, \end{aligned}$$

where it was used that $\frac{q_\beta(\boldsymbol{\beta}^{(l)} \mid \boldsymbol{\beta}^*)}{q_\beta(\boldsymbol{\beta}^* \mid \boldsymbol{\beta}^{(l)})} = 1$ and $\frac{q_{\sigma^2}(\sigma^{2(l)} \mid \sigma^{2*})}{q_{\sigma^2}(\sigma^{2*} \mid \sigma^{2(l)})} = \frac{\sigma^{2*}}{\sigma^{2(l)}}$. Then set

$$(\boldsymbol{\beta}^{(l+1)}, \sigma^{2(l+1)}) = \begin{cases} (\boldsymbol{\beta}^*, \sigma^{2*}) & \text{if } U_1 < \alpha_1 \\ (\boldsymbol{\beta}^{(l)}, \sigma^{2(l)}) & \text{otherwise} \end{cases}.$$

Step 3: Simulate independently $\xi_3^* \sim \mathrm{N}(\xi_3^{(l)}, c_3)$, $\xi_4^* \sim \mathrm{N}(\xi_4^{(l)}, c_4)$ and $U_2 \sim \text{unif}(0, 1)$. The latent process is updated via the Langevin–Hastings algorithm which uses gradient information of the target density (Christensen et al., 2006). For k, $k' = 1, \ldots, K$, the proposed k^{th} set of random effects is simulated independently of the k'^{th} set ($k \neq k'$) as then set $\boldsymbol{\gamma}_{(K)}^* = (\boldsymbol{\gamma}_1^*, \ldots, \boldsymbol{\gamma}_K^*)$.

$$\boldsymbol{\gamma}_k^* \sim \mathrm{N}_n\left(\boldsymbol{\gamma}_k^{(l)} + \frac{c_5}{2} \nabla \log \pi(\boldsymbol{\gamma}_k^{(l)} \mid \mathbf{y}, \boldsymbol{\beta}^{(l+1)}, \sigma^{2(l+1)}, \theta^{(l)}, \tau^{2(l)}), c_5 I_n\right);$$

Step 4: Compute the acceptance probability $\alpha_2 = \min\{1, t_2\}$, where

$$\begin{aligned} t_2 &= \frac{\pi_{(K)}(\theta^*, \tau^{2*}, \boldsymbol{\gamma}_{(K)}^* \mid \boldsymbol{\beta}^{(l)}, \sigma^{2(l)}, \mathbf{y}_{(K)}) q_\theta(\theta^{(l)} \mid \theta^*) q_{\tau^2}(\tau^{2(l)} \mid \tau^{2*}) q_{\gamma_{(K)}}(\boldsymbol{\gamma}_{(K)}^{(l)} \mid \boldsymbol{\gamma}_{(K)}^*)}{\pi_{(K)}(\theta^{(l)}, \tau^{2(l)}, \boldsymbol{\gamma}_{(K)}^{(l)} \mid \boldsymbol{\beta}^{(l)}, \sigma^{2(l)}, \mathbf{y}_{(K)}) q_\theta(\theta^* \mid \theta^{(l)}) q_{\tau^2}(\tau^{2*} \mid \tau^{2(l)}) q_{\boldsymbol{\gamma}_{(K)}}(\boldsymbol{\gamma}_{(K)}^* \mid \boldsymbol{\gamma}_{(K)}^{(l)})} \\ &= \frac{\pi(\theta^*)\pi(\tau^{2*})\prod_{k=1}^K \phi_n(\boldsymbol{\gamma}_k^*; (1-\tau^{2*})R_{\theta^*})\prod_{k=1}^K \prod_{i=1}^n h_i(y_i \mid \gamma_{ki}^*; \boldsymbol{\beta}^{(l)}, \sigma^{2(l)}, \tau^{2*}) q_{\gamma_{(K)}}(\boldsymbol{\gamma}_{(K)}^{(l)} \mid \boldsymbol{\gamma}_{(K)}^*)\theta^*\tau^{2*}(1-\tau^{2*})}{\pi(\theta^{(l)})\pi(\tau^{2(l)})\prod_{k=1}^K \phi_n(\boldsymbol{\gamma}_k^*; (1-\tau^{2(l)})R_{\theta^{(l)}})\prod_{k=1}^K \prod_{i=1}^n h_i(y_i \mid \gamma_{ki}^{(l)}; \boldsymbol{\beta}^{(l)}, \sigma^{2(l)}, \tau^{2(l)}) q_{\gamma_{(K)}}(\boldsymbol{\gamma}_{(K)}^* \mid \boldsymbol{\gamma}_{(K)}^{(l)})\theta^{(l)}\tau^{2(l)}(1-\tau^{2(l)})} \end{aligned}$$

where

$$\begin{aligned} q_{\gamma(K)}(\boldsymbol{\gamma}^*_{(K)} \mid \boldsymbol{\gamma}^{(l)}_{(K)}) &= \prod_{k=1}^{K} q_{\gamma_k}(\boldsymbol{\gamma}^*_k \mid \boldsymbol{\gamma}^{(l)}_k) \\ &= \prod_{k=1}^{K} \phi_n\left(\boldsymbol{\gamma}^*_k - \boldsymbol{\gamma}^{(l)}_k - \frac{c_5}{2} \nabla \log \pi(\boldsymbol{\gamma}^{(l)}_k \mid \boldsymbol{y}, \boldsymbol{\beta}^{(l+1)}, \sigma^{2(l+1)}, \theta^{(l)}, \tau^{2(l)}); c_5 I_n\right), \end{aligned}$$

$q_{\gamma(K)}(\boldsymbol{\gamma}^{(l)}_{(K)} \mid \boldsymbol{\gamma}^*_{(K)})$ is similarly defined, and it was used that $\frac{q_\theta(\theta^{(l)} \mid \theta^*)}{q_\theta(\theta^* \mid \theta^{(l)})} = \frac{\theta^*}{\theta^{(l)}}$ and $\frac{q_{\tau^2}(\tau^{2(l)} \mid \tau^{2\,*})}{q_{\tau^2}(\tau^{2\,*} \mid \tau^{2(l)})} = \frac{\tau^{2\,*}(1 - \tau^{2\,*})}{\tau^{2(l)}(1 - \tau^{2(l)})}$. Then set

$$(\theta^{(l+1)}, \tau^{2(l+1)}, \boldsymbol{\gamma}^{(l+1)}_{(K)}) = \begin{cases} (\theta^*, \tau^{2\,*}, \boldsymbol{\gamma}^*_{(K)}) & \text{if } U_2 < \alpha_2 \\ (\theta^{(l)}, \tau^{2(l)}, \boldsymbol{\gamma}^{(l)}_{(K)}) & \text{otherwise} \end{cases} .$$

Step 5: Increase l by 1, and repeat steps 1–4 until the states have reached the equilibrium distribution of the Markov chain.

Posterior Calculation of the Gradient of the Latent Process

The log of the full conditional distribution of the latent field γ is given by

$$\log(\pi(\boldsymbol{\gamma} \mid \boldsymbol{y}, \boldsymbol{\psi}, \theta, \tau^2)) = \log(\phi_n(\boldsymbol{\gamma}; (1 - \tau^2)R_\theta)) + \sum_{i=1}^{n} \log(h_i(y_i \mid \gamma_i; \boldsymbol{\psi}, \tau^2)) + \text{const}\,.$$

For $i = 1,\ldots,n$ we have

$$\frac{\partial}{\partial \gamma_i} \log(h_i(y_i \mid \gamma_i; \boldsymbol{\psi}, \tau^2)) = \frac{\phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y_i - 1; \boldsymbol{\psi})) - \gamma_i}{\tau}\right) - \phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y_i; \boldsymbol{\psi})) - \gamma_i}{\tau}\right)}{\tau\left[\Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y_i; \boldsymbol{\psi})) - \gamma_i}{\tau}\right) - \Phi\left(\frac{\Phi^{-1}(F_{\mathbf{s}_i}(y_i - 1; \boldsymbol{\psi})) - \gamma_i}{\tau}\right)\right]}, \tag{13}$$

and since $\log(\phi_n(\boldsymbol{\gamma}; (1 - \tau^2)R_\theta)) = -\frac{1}{2}(\log|(1 - \tau^2)R_\theta| + (1 - \tau^2)^{-1}\boldsymbol{\gamma}^\intercal R_\theta^{-1}\boldsymbol{\gamma} + n\log(2\pi)),$

$$\frac{\partial}{\partial \boldsymbol{\gamma}} \log(\phi_n(\boldsymbol{\gamma}; (1 - \tau^2)R_\theta)) = -\frac{R_\theta^{-1}\boldsymbol{\gamma}}{1 - \tau^2}\,. \tag{14}$$

Combining (13) and (14) we get ∇ log ($\boldsymbol{\pi}(\gamma \mid \gamma, \mid \mathbf{y}, \boldsymbol{\psi}, \boldsymbol{\theta}, \tau^2)$).

Remark 4. The tuning constants $c_1,\ldots,c_4$ could be selected by trial and error until both Metropolis–Hastings steps have good acceptance probabilities (in the range 0.25–0.5, say). Instead, in the simulation we used the algorithm in Haario et al. (1999) that automatically adjusts these constants every so often. Let $\xi_1^{(l)}, \xi_2^{(l)}, \xi_3^{(l)}, \xi_4^{(l)}$ be the transformed parameters at the l^{th} iteration. The variances of their proposal distributions are adjusted every 1000 iterations, at iterations $l \in \{1001, 2001, \dots\}$, as

$q_{\xi_1}(\cdot \mid \xi_1^{(l-1000)}, \dots, \xi_1^{(l-1)}) \sim \mathrm{N}(\xi_1^{(l-1)}, r_{d_1}^2 \widehat{\mathrm{var}}(\xi_1^{(l)}))$,

$q_{\xi_2}(\cdot \mid \xi_2^{(l-1000)}, \dots, \xi_2^{(l-1)}) \sim \mathrm{N}(\xi_2^{(l-1)}, r_{d_1}^2 \widehat{\mathrm{var}}(\xi_2^{(l)}))$,

$q_{\xi_3}(\cdot \mid \xi_3^{(l-1000)}, \dots, \xi_3^{(l-1)}) \sim \mathrm{N}(\xi_3^{(l-1)}, r_{d_2}^2 \widehat{\mathrm{var}}(\xi_3^{(l)}))$ and

$q_{\xi_4}(\cdot \mid \xi_4^{(l-1000)}, \dots, \xi_4^{(l-1)}) \sim \mathrm{N}(\xi_4^{(l-1)}, r_{d_2}^2 \widehat{\mathrm{var}}(\xi_4^{(l)}))$, where $\widehat{\mathrm{var}}(\xi_1^{(l)})$, $\widehat{\mathrm{var}}(\xi_2^{(l)})$, $\widehat{\mathrm{var}}(\xi_3^{(l)})$ and $\widehat{\mathrm{var}}(\xi_4^{(l)})$ are the sample variance matrices computed from their previous 1000 iterations, $r_{d_1} = 2.38 / \sqrt{d_1}$ and $r_{d_2} = 2.38 / \sqrt{d_2}$, with d_1 and d_2 the dimensions of $(\xi_1^{(l)}, \xi_2^{(l)})$ and $(\xi_3^{(l)}, \xi_4^{(l)})$.

References

Bai Y, Kang J and Song P-X (2014). Efficient Pairwise Composite Likelihood Estimation for Spatial–Clustered Data, Biometrics, 70: 661–670. [PubMed: 24945876]

Baghishani H, Rue H and Mohammadzadeh M (2012). On a Hybrid Data Cloning Method and Its Applications in Generalized Linear Mixed Models, Statistics and Computing, 22: 597–613.

Baghishani H and Mohammadzadeh M (2011). A Data Cloning Algorithm for Computing Maximum Likelihood Estimates in Spatial Generalized Linear Mixed Models, Computational Statistics and Data Analysis, 55: 1748–1759.

Cameron AC and Trivedi PK (2013). Regression Analysis of Count Data, 2nd ed., Cambridge University Press.

Christensen OF, Roberts GO and Sköld M (2006). Robust Markov Chain Monte Carlo Methods for Spatial Generalized Linear Mixed Models, Journal of Computational and Graphical Statistics, 15: 1–17.

De Oliveira V (2013). Hierarchical Poisson Models for Spatial Count Data, Journal of Multivariate Analysis, 122: 393–408.

Diggle PJ, Tawn JA and Moyeed RA (1998). Model–Based Geostatistics (with discussion), Journal of the Royal Statistical Society, Series C, 47: 299–326.

Eddelbuettel D and Sanderson C (2014). RcppArmadillo: Accelerating R with High–Performance C++ Linear Algebra, Computational Statistics and Data Analysis, 71: 1054–1063.

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

Genz A and Bretz F (2009). Computation of Multivariate Normal and t Probabilities, Springer.

Genz A and Bretz F (2002). Comparison of Methods for the Computation of Multivariate t–Probabilities, Journal of Computational and Graphical Statistics, 11: 950–971.

Geweke J (1991). Efficient Simulation from the Multivariate Normal and Student–*t* Distributions Subject to Linear Constraints. Computer Science and Statistics: Proceedings of the Twenty Third Symposium on the Interface: 571–578.

Haario H, Saksman E and Tamminen J (1999). Adaptive Proposal Distribution for Random Walk Metropolis Algorithm, Computational Statistics, 14: 375–395.

Hajivassiliou V, McFadden D and Ruud P (1996). Simulation of Multivariate Normal Rectangle Probabilities and Their Derivatives: Theoretical and Computational Results, Journal of Econometrics, 72: 85–134.

Han Z and De Oliveira V (2018). gcKrig: An R Package for the Analysis of Geostatistical Count Data Using Gaussian Copulas, Journal of Statistical Software, to appear.

Han Z and De Oliveira V (2016). On the Correlation Structure of Gaussian Copula Models for Geostatistical Count Data, Australian and New Zealand Journal of Statistics, 58: 47–69.

Irvine K, Gitelman AI and Hoeting JA (2007). Spatial Designs and Properties of Spatial Correlation: Effects on Covariance Estimation, Journal of Agricultural, Biological, and Environmental Statistics, 12: 450–469.

Hickernell FJ (1998). A Generalized Discrepancy and Quadrature Error Bound, Mathematics of Computation, 67: 299–322.

Kazianka H (2013). Approximate Copula–Based Estimation and Prediction of Discrete Spatial Data, Stochastic Environmental Research and Risk Assessment, 27: 2015–2026.

Kazianka H and Pilz J (2010). Copula-Based Geostatistical Modeling of Continuous and Discrete Data Including Covariates, Stochastic Environmental Research and Risk Assessment, 24: 661–673.

Keane MP (1994). A Computationally Practical Simulation Estimator for Panel Data, Econometrica, 62: 95–116.

Knaus J (2015). snowfall: Easier Cluster Computing, R package version 1.84–6.1, http://cran.r-project.org/package=snowfall.

Lele SR, Nadeem K and Schmuland B (2010). Estimability and Likelihood Inference for Generalized Linear Mixed Models Using Data Cloning, Journal of the American Statistical Association, 105: 1617–1625.

Lele SR, Dennis B and Lutscher F (2007). Data Cloning: Easy Maximum Likelihood Estimation for Complex Ecological Models Using Bayesian Markov Chain Monte Carlo Methods, Ecology Letters, 10: 551–563. [PubMed: 17542934]

Madsen L (2009). Maximum Likelihood Estimation of Regression Parameters With Spatially Dependent Discrete Data, Journal of Agricultural, Biological, and Environmental Statistics, 14: 375–391.

Madsen L and Fang Y (2011). Joint Regression Analysis for Discrete Longitudinal Data, Biometrics, 67: 1171–1176. [PubMed: 21039391]

Masarotto G and Varin C (2017). Gaussian Copula Regression in R, Journal of Statistical Software, 77(8): 1–26.

Masarotto G and Varin C (2012). Gaussian Copula Marginal Regression, Electronic Journal of Statistics, 6: 1517–1549.

Meeker WQ and Escobar LA (1995). Teaching About Approximate Confidence Regions Based on Maximum Likelihood Estimation, The American Statistician, 49: 48–53.

Nikoloulopoulos AK (2016). Efficient Estimation of High–Dimensional Multivariate Normal Copula Models with Discrete Spatial Responses, Stochastic Environmental Research and Risk Assessment, 30: 493–505.

Nikoloulopoulos AK (2013). On the Estimation of Normal Copula Discrete Regression Models Using the Continuous Extension and Simulated Likelihood, Journal of Statistical Planning and Inference, 143: 1923–1937.

Ponciano JM, Taper ML, Dennis B and Lele SR (2009). Hierarchical Models in Ecology: Confidence Intervals, Hypothesis Testing, and Model Selection Using Data Cloning, Ecology, 90: 356–362. [PubMed: 19323219]

Sloan IH and Kachoyan PJ (1987). Lattice Methods for Multiple Integration: Theory, Error Analysis and Examples, SIAM Journal on Numerical Analysis, 24: 116–128.

Song PX (2000). Multivariate Dispersion Models Generated From Gaussian Copula, Scandinavian Journal of Statistics, 27: 305–320.

Thompson EA (1994). Monte Carlo Likelihoods in Genetic Mapping, Statistical Science, 9: 355–366.

Torabi M (2015). Likelihood Inference for Spatial Generalized Linear Mixed Models, Communications in Statistics—Simulation and Computation, 44: 1692–1701.

Varin C, Reid N and Firth D (2011). An Overview of Composite Likelihood Methods, Statistica Sinica, 21: 5–42.

Walker AM (1969). On the Asymptotic Behavior of Posterior Distributions, Journal of the Royal Statistical Society B, 31: 80–88.

Zhao Y and Joe H (2005). Composite Likelihood Estimation in Multivariate Data Analysis, The Canadian Journal of Statistics, 33: 335–356.

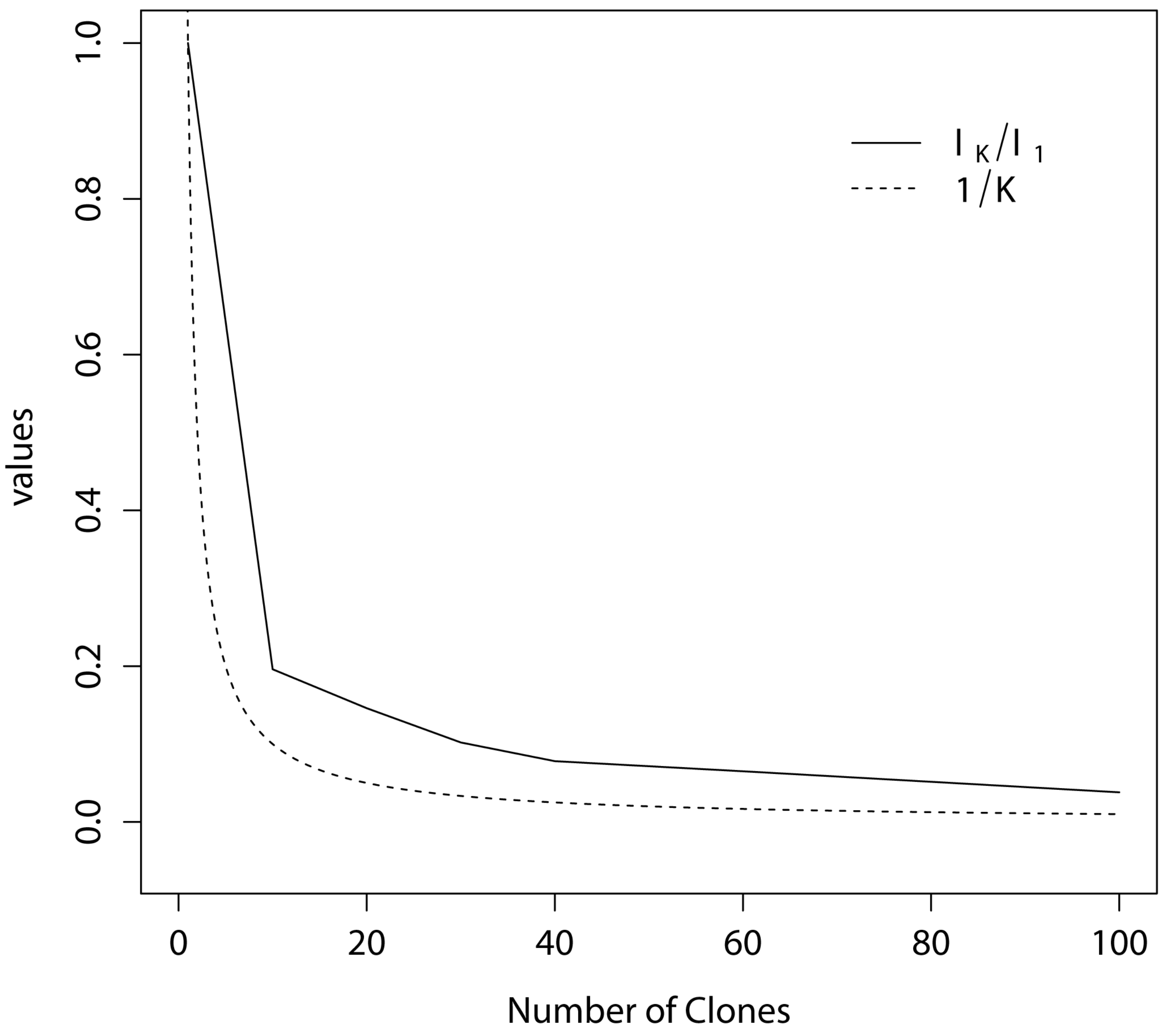


Figure 1:

Plots of standardized eigenvalues of the posterior variance matrix based on the number of clones K (solid line), and the function $1/K$ (dashed line).

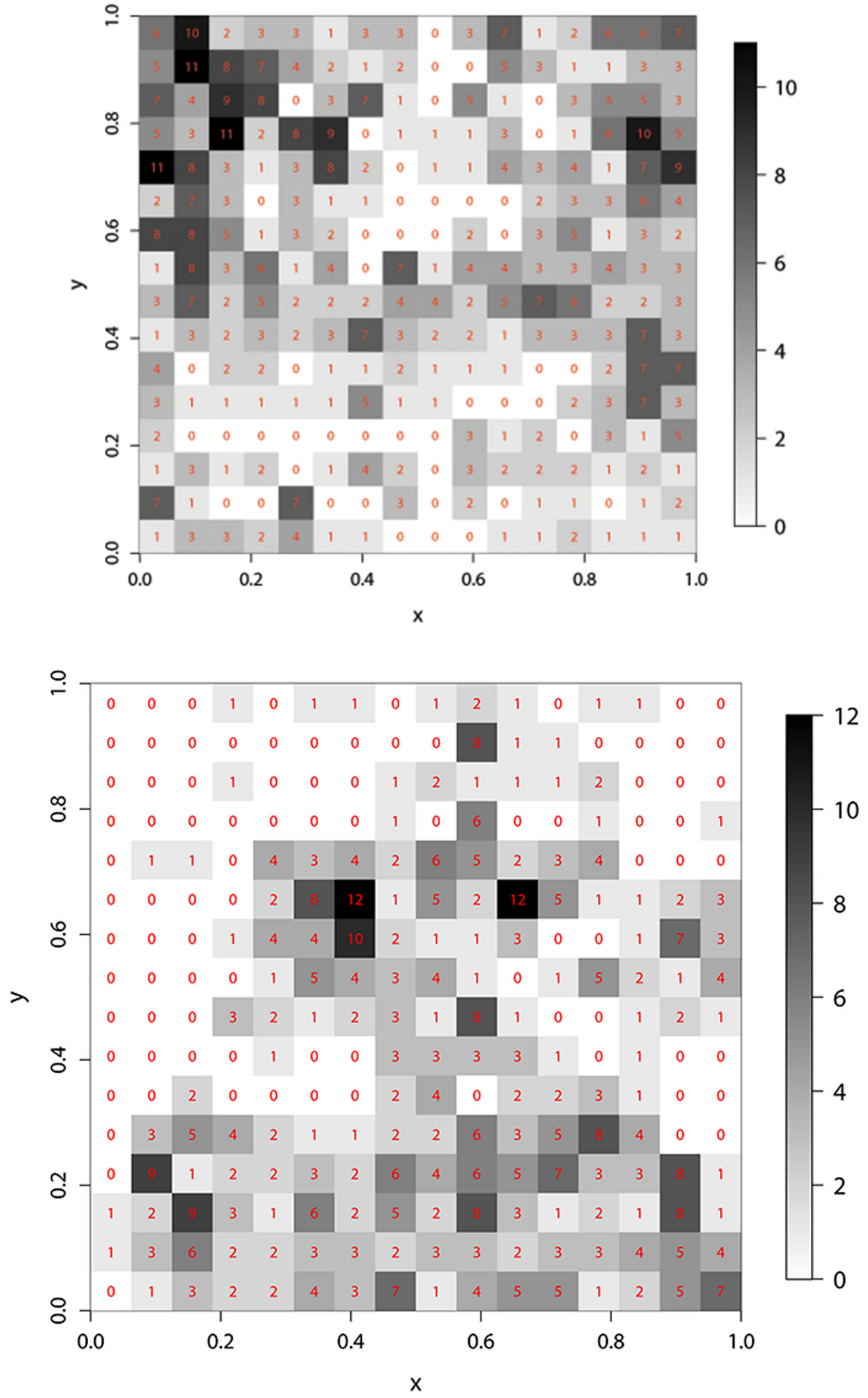
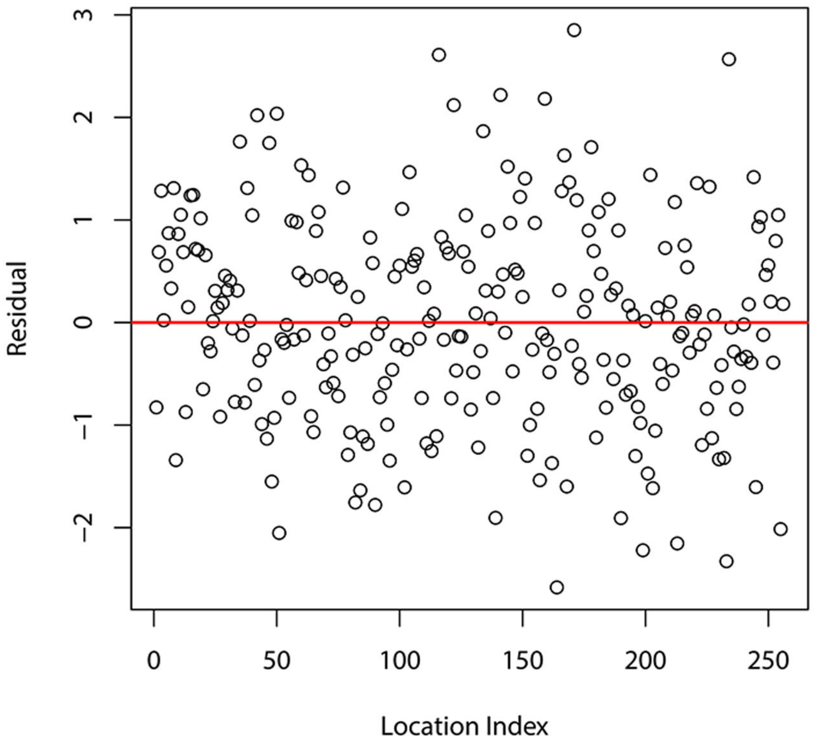
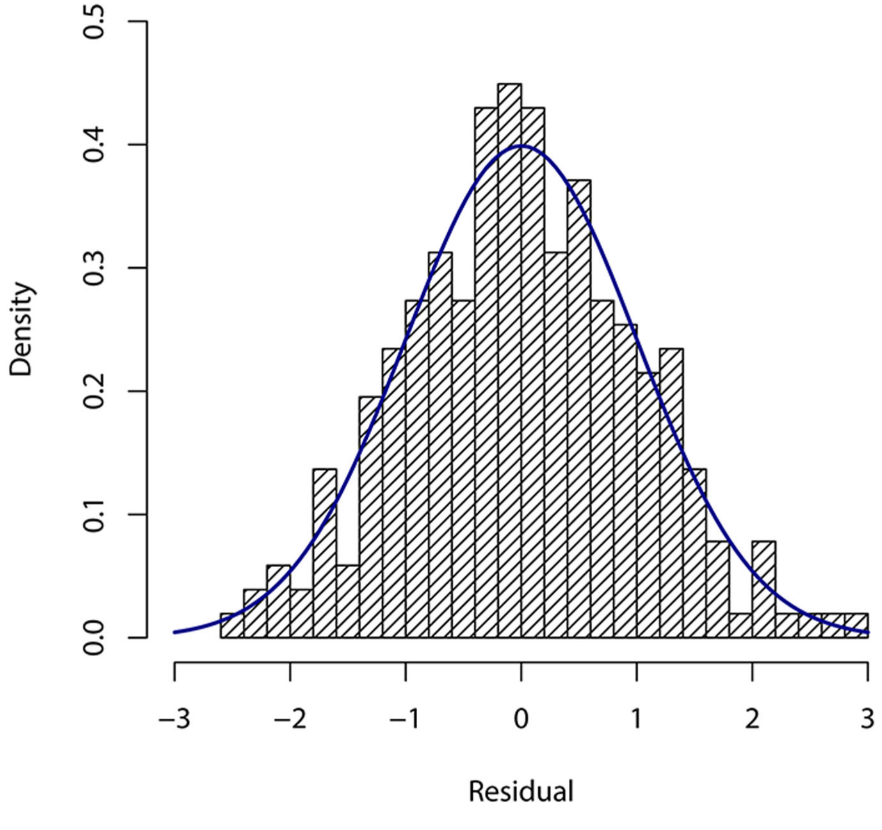


Figure 2:

Lansing wood dataset: spatial distribution of hickory counts (left) and maple counts (right).


Residual
3
2
1
0
−1
−2
0
50
100
150
200
250
Location Index


Density
0.5
0.4
0.3
0.2
0.1
0.0
−3
−2
−1
0
1
2
3
Residual

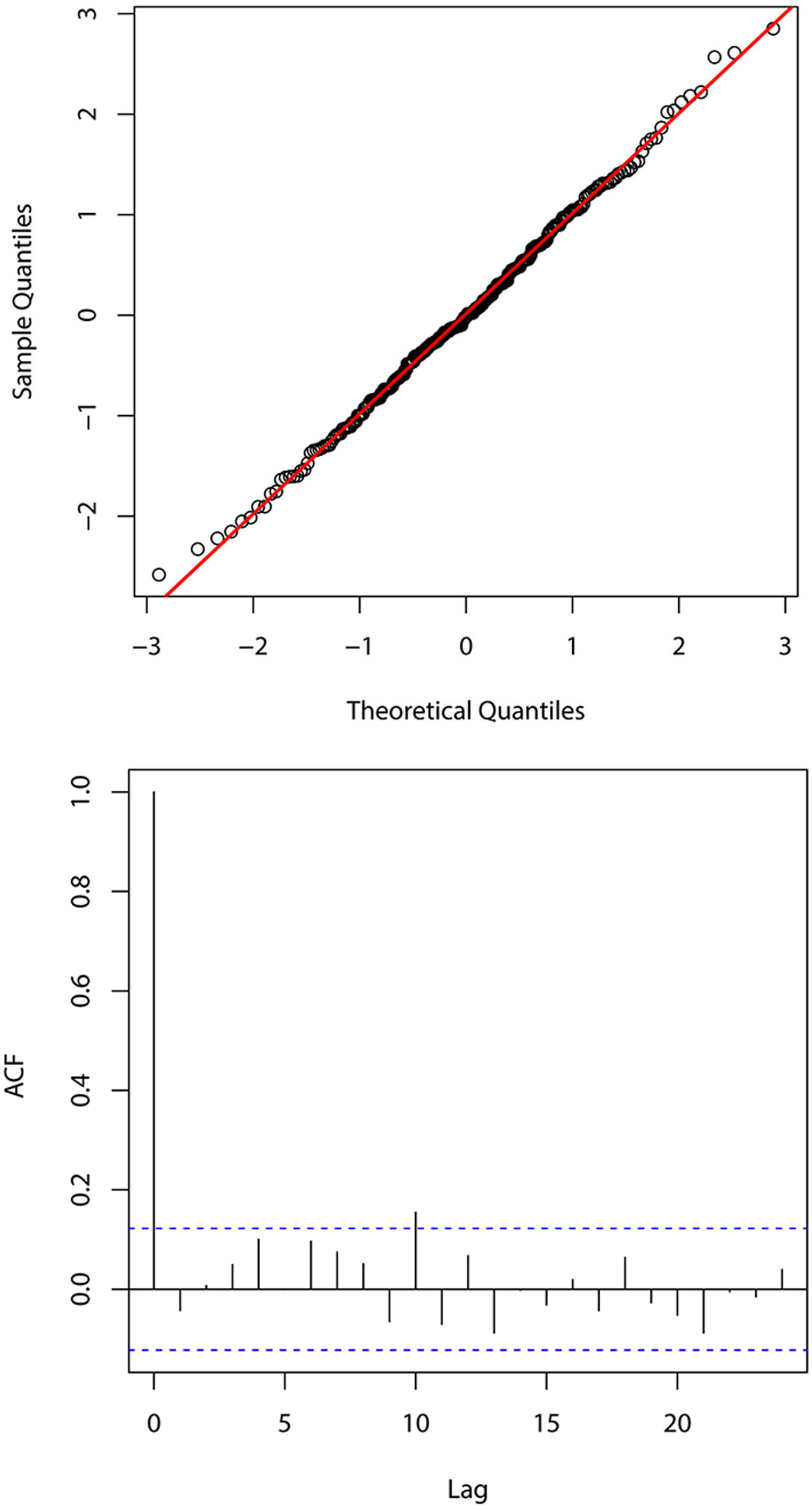


Figure 3:

Graphical summaries of residuals from the fitted Gaussian copula model with negative binomial marginals.

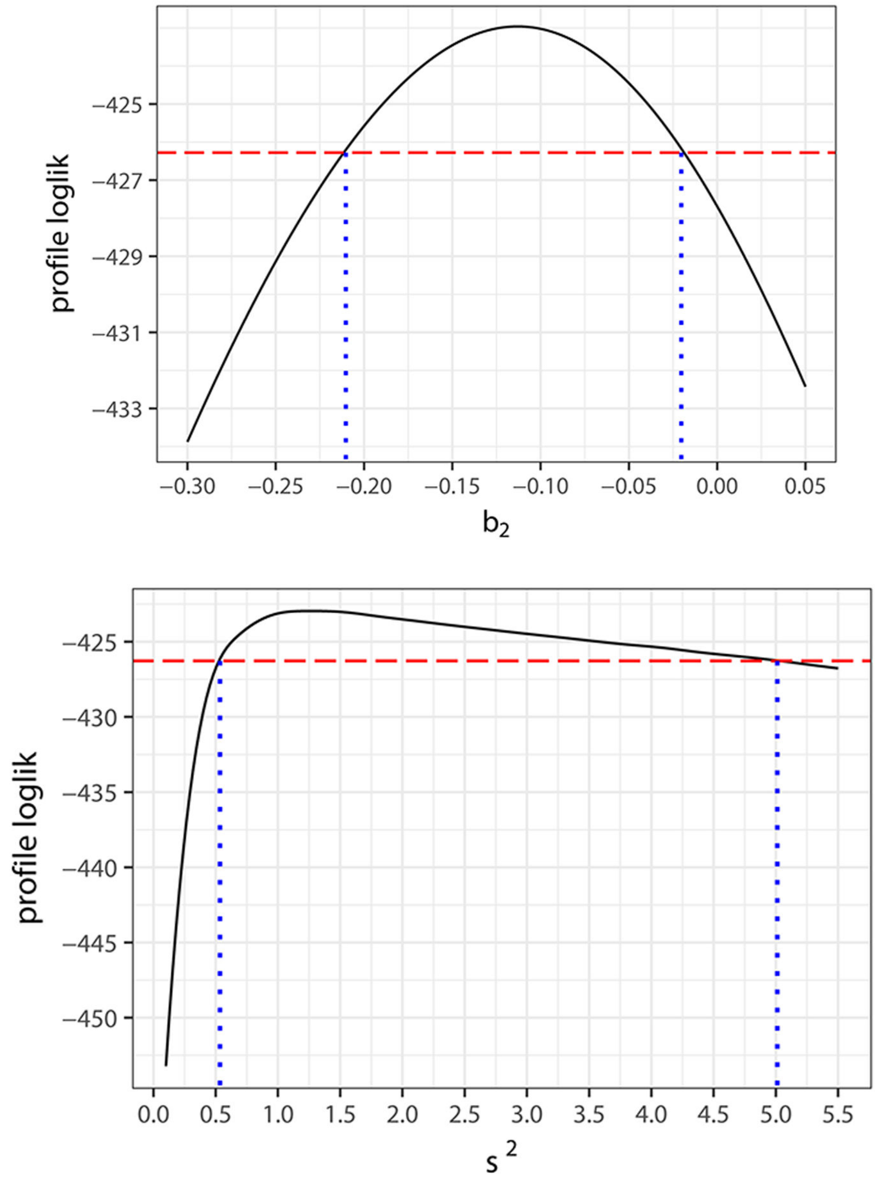


Figure 4:

Profile likelihoods of the parameter β_2 and σ^2 under the NB2 marginal and their corresponding 99% confidence intervals.

Table 1:

Absolute relative bias (ARB) and mean absolute errors (MAE) for the three methods (GB, GHK and DC) and four simulation scenarios. The summaries are based on 1000 simulated datasets.

True parameters	ARB			MAE		
	GB	GHK	DC	GB	GHK	DC
$\beta_1 = 0.5$	0.06	0.06	0.14	0.25	0.24	0.26
$\beta_2 = 0.5$	0.00	0.00	0.02	0.28	0.28	0.29
$\beta_3 = 1.0$	0.02	0.01	0.00	0.28	0.28	0.29
$\sigma^2 = 0.2$	0.15	0.15	0.10	0.07	0.07	0.08
$\theta = 0.1$	0.10	0.20	0.20	0.04	0.04	0.04
$\tau^2 = 0.25$	0.40	0.36	0.32	0.22	0.21	0.20
$\beta_1 = 0.5$	0.30	0.26	0.30	0.44	0.43	0.44
$\beta_2 = 0.5$	0.10	0.10	0.04	0.44	0.45	0.45
$\beta_3 = 1.0$	0.07	0.05	0.02	0.44	0.44	0.47
$\sigma^2 = 0.2$	0.50	0.55	0.35	0.13	0.14	0.15
$\theta = 0.3$	0.40	0.43	0.33	0.15	0.15	0.13
$\tau^2 = 0.25$	0.28	0.24	0.24	0.16	0.15	0.15
$\beta_1 = 0.5$	0.18	0.16	0.12	0.46	0.45	0.46
$\beta_2 = 0.5$	0.02	0.02	0.06	0.54	0.54	0.58
$\beta_3 = 1.0$	0.03	0.03	0.08	0.55	0.55	0.60
$\sigma^2 = 2.0$	0.06	0.05	0.04	0.37	0.38	0.40
$\theta = 0.1$	0.10	0.10	0.10	0.04	0.04	0.04
$\tau^2 = 0.25$	0.32	0.32	0.28	0.21	0.22	0.22
$\beta_1 = 0.5$	0.64	0.56	0.58	0.75	0.74	0.73
$\beta_2 = 0.5$	0.10	0.06	0.02	0.78	0.77	0.80
$\beta_3 = 1.0$	0.06	0.05	0.12	0.76	0.77	0.82
$\sigma^2 = 2.0$	0.14	0.14	0.09	0.71	0.71	0.70
$\theta = 0.3$	0.33	0.33	0.30	0.17	0.16	0.16
$\tau^2 = 0.25$	0.24	0.24	0.20	0.17	0.17	0.18

Table 2:

Empirical coverage probability and median length of different nominal 95% confidence intervals using the GHK method. The summaries are based on 1000 simulated datasets.

True parameters	Coverage			Length		
	Wald	Wald–Tran	PL	Wald	Wald–Tran	PL
$\beta_1 = 0.5$	0.902	0.900	0.927	0.977	0.974	1.076
$\beta_2 = 0.5$	0.904	0.901	0.918	1.135	1.131	1.232
$\beta_3 = 1.0$	0.908	0.907	0.929	1.142	1.140	1.243
$\sigma^2 = 0.2$	0.875	0.979	0.972	0.268	0.500	0.355
$\theta = 0.1$	0.971	0.767	0.941	0.291	0.318	0.348
$\tau^2 = 0.25$	0.968	0.954	0.952	0.901	0.912	0.735
$\beta_1 = 0.5$	0.765	0.759	0.831	1.238	1.221	1.439
$\beta_2 = 0.5$	0.787	0.782	0.842	1.349	1.335	1.521
$\beta_3 = 1.0$	0.789	0.788	0.839	1.383	1.362	1.542
$\sigma^2 = 0.2$	0.504	0.822	0.796	0.228	0.502	0.353
$\theta = 0.3$	0.677	0.701	0.929	0.360	0.435	0.678
$\tau^2 = 0.25$	0.953	0.948	0.957	0.675	0.726	0.612
$\beta_1 = 0.5$	0.901	0.899	0.928	1.866	1.859	2.001
$\beta_2 = 0.5$	0.906	0.904	0.941	2.309	2.300	2.432
$\beta_3 = 1.0$	0.912	0.912	0.939	2.329	2.321	2.446
$\sigma^2 = 2.0$	0.859	0.897	0.868	1.651	1.678	1.592
$\theta = 0.1$	0.986	0.785	0.956	0.318	0.385	0.427
$\tau^2 = 0.25$	0.971	0.956	0.961	0.886	0.891	0.769
$\beta_1 = 0.5$	0.782	0.776	0.881	2.293	2.284	2.597
$\beta_2 = 0.5$	0.846	0.839	0.896	2.711	2.689	2.869
$\beta_3 = 1.0$	0.847	0.844	0.894	2.694	2.685	2.834
$\sigma^2 = 2.0$	0.696	0.807	0.804	2.001	2.140	1.964
$\theta = 0.3$	0.698	0.763	0.962	0.414	0.572	0.809
$\tau^2 = 0.25$	0.949	0.971	0.973	0.690	0.715	0.633

Table 3:

ARB, MAE, empirical coverage probability and median length of nominal 95% confidence intervals using the GHK method for the second simulation study. The summaries are based on 1000 simulated datasets.

True parameters	GHK Estimate		Coverage		Length	
	ARB	MAE	Wald–Tran	PL	Wald–Tran	PL
$\beta_1 = 0.5$	0.02	0.12	0.933	0.938	0.556	0.562
$\beta_2 = 0.5$	0.00	0.07	0.930	0.935	0.317	0.323
$\beta_3 = 1.0$	0.01	0.07	0.918	0.932	0.325	0.348
$\sigma^2 = 0.2$	0.05	0.02	0.950	0.951	0.107	0.108
$\theta = 0.1$	0.10	0.02	0.784	0.967	0.120	0.126
$\tau^2 = 0.25$	0.24	0.16	0.944	0.980	0.798	0.539
$\beta_1 = 0.5$	0.14	0.24	0.882	0.903	0.953	1.019
$\beta_2 = 0.5$	0.02	0.12	0.897	0.915	0.504	0.534
$\beta_3 = 1.0$	0.03	0.13	0.880	0.923	0.525	0.584
$\sigma^2 = 0.2$	0.10	0.05	0.844	0.893	0.173	0.175
$\theta = 0.3$	0.13	0.09	0.849	0.924	0.323	0.384
$\tau^2 = 0.25$	0.00	0.08	0.964	0.976	0.296	0.310
$\beta_1 = 0.5$	0.06	0.23	0.931	0.940	1.082	1.103
$\beta_2 = 0.5$	0.00	0.14	0.928	0.934	0.662	0.676
$\beta_3 = 1.0$	0.00	0.14	0.945	0.951	0.666	0.672
$\sigma^2 = 2.0$	0.01	0.17	0.941	0.929	1.531	1.417
$\theta = 0.1$	0.00	0.02	0.789	0.974	0.123	0.125
$\tau^2 = 0.25$	0.16	0.17	0.934	0.972	0.722	0.552
$\beta_1 = 0.5$	0.12	0.36	0.931	0.938	1.541	1.610
$\beta_2 = 0.5$	0.00	0.17	0.944	0.948	0.817	0.855
$\beta_3 = 1.0$	0.00	0.18	0.941	0.945	0.828	0.878
$\sigma^2 = 2.0$	0.02	0.37	0.937	0.918	1.124	1.093
$\theta = 0.3$	0.07	0.09	0.893	0.936	0.360	0.433
$\tau^2 = 0.25$	0.00	0.07	0.966	0.946	0.315	0.312

Table 4:

Computational time (in seconds) needed to compute the average MLE for the three methods from 100 simulations with different sample sizes using 4 cores.

n	**25**	**64**	**121**	**256**	**441**	**961**
GB	10.05	18.37	62.23	116.21	286.81	1060.21
GHK	0.38	0.50	1.80	7.88	35.87	108.60
DC	28.20	37.38	69.47	200.17	367.62	1748.41

Table 5:

MLE and their estimated asymptotic standard deviations (in parentheses) of the GHK and DC methods from the Lansing Woods dataset, assuming Poisson marginals. GQT and CL estimates are from Kazianka (2013).

Method	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\theta}$	$\hat{\tau}^2$
GHK	1.124 (0.097)	−0.152 (0.027)	0.084 (0.023)	0.504 (0.127)
DC	1.123 (0.113)	−0.150 (0.025)	0.083 (0.018)	0.494 (0.101)
GQT	1.057	−0.181	0.063	0.470
CL	1.227	−0.253	0.060	0.421

Table 6:

MLE and their asymptotic standard errors (in parentheses) for the Lansing Woods dataset, obtained from the GHK and DC methods, assuming ZIP and NB2 marginals.

	Method	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\sigma}^2$	$\hat{\theta}$	$\hat{\tau}^2$
ZIP	GHK	0.905 (0.146)	−0.155 (0.030)	0.357 (0.119)	0.084 (0.024)	0.414 (0.149)
	DC	0.895 (0.131)	−0.155 (0.026)	0.355 (0.108)	0.082 (0.021)	0.408 (0.116)
NB2	GHK	0.747 (0.319)	−0.112 (0.037)	1.278 (0.484)	0.172 (0.081)	0.208 (0.097)
	DC	0.742 (0.292)	−0.113 (0.038)	1.264 (0.439)	0.178 (0.076)	0.212 (0.101)

Table 7:

Model comparison summaries from the Lansing Woods dataset for Gaussian copula models with different marginals.

Model	Number of Parameters	Log–likelihood	AIC$_c$	BIC
Poisson	4	−477.16	962.48	976.50
ZIP	5	−454.71	919.66	937.15
NB2	5	−422.96	856.16	873.65