Estimating orthant probabilities of high dimensional Gaussian vectors with an application to set estimation

Dario Azzimonti, David Ginsbourger

Estimating orthant probabilities of high dimensional Gaussian vectors with an application to set estimation

Dario Azzimonti [*‡] David Ginsbourger [†*]

Abstract

The computation of Gaussian orthant probabilities has been extensively studied for low dimensional vectors. Here we focus on the high dimensional case and we present a two step procedure relying on both deterministic and stochastic techniques. The proposed estimator relies indeed on splitting the probability into a low dimensional term and a remainder. While the low dimensional probability can be estimated by fast and accurate quadrature, the remainder requires Monte Carlo sampling. We further refine the estimation by using a novel asymmetric nested Monte Carlo (anMC) algorithm for the remainder and we highlight cases where this approximation brings substantial efficiency gains. The proposed methods are compared against state-of-the-art methods in a numerical study, where we also analyse advantages and drawbacks of anMC. Finally the proposed method is applied to derive conservative estimates of excursion sets of expensive to evaluate deterministic functions under a Gaussian random field prior, without requiring a Markov assumption.

Keywords: Conservative set estimation; Gaussian probabilities; Gaussian random fields; Monte Carlo.

[*]IMSV, Department of Mathematics and Statistics, University of Bern, Alpeneggstrasse 22, 3012 Bern, Switzerland

[†]Uncertainty Quantification and Optimal Design group, Idiap Research Institute, Centre du Parc, Rue Marconi 19, PO Box 592, 1920 Martigny, Switzerland.

[‡]The first author gratefully acknowledges the support of the *Swiss National Science Foundation, grant number 146354.*

1 Introduction

Assume that $X = (X_1, \ldots, X_d)$ is a random vector with Gaussian distribution $N_d(\mu, \Sigma)$. We are interested in estimating, for any fixed $t \in \mathbb{R}$, the following probability

$$\pi(t) = P(X \leq (t, \ldots, t)). \tag{1}$$

The general problem of evaluating $\pi(t)$, which, for a full rank matrix Σ, is the integral of the multivariate normal density $\phi(\cdot; \mu, \Sigma)$ over the one-sided d-dimensional rectangle $(-\infty, t]^d$, has been extensively studied in moderate dimensions with many different methods. In low dimensions tables are available (see, e.g., Owen (1956) for $d = 2$). Furthermore, when the dimension is smaller than 20, there exist methods (see, e.g., Abrahamson (1964), Moran (1984) and Miwa et al. (2003)) exploiting the specific orthant structure of the probability in (1). Currently, however, most of the literature uses numerical integration techniques to approximate the quantity. In moderate dimensions fast reliable methods are established to approximate $\pi(t)$ (see, e.g. Cox and Wermuth (1991)) and more recently the methods introduced in Schervish (1984); Genz (1992) and Hajivassiliou et al. (1996) (see also Genz and Bretz (2002), Ridgway (2016) and the book Genz and Bretz (2009) for a broader overview) provide state-of-the-art algorithms when $d < 100$. The method introduced by Genz (1992) has been recently revised in Botev (2017) where a more efficient tilted estimator is proposed. Those techniques rely on fast quasi Monte Carlo (qMC) methods and are very accurate for moderate dimensions. However, when d is larger than 1000 they are not computationally efficient or become intractable. Commonly used alternative methods are standard Monte Carlo (MC) techniques (see Tong (2012), Chapter 8 for an extensive review), for which getting accurate estimates can be computationally prohibitive.

We propose here a two step method that exploits the power of qMC quadratures and the flexibility of stochastic simulation for the specific problem of estimating $\pi(t)$. We rely on the following equivalent formulation.

$$\pi(t) = 1 - P(\max X > t),$$

where $\max X$ denotes $\max_{i=1,\ldots,d} X_i$. In the following we fix t and denote $p = P(\max X > t)$.

The central idea is using a moderate dimensional subvector of X to approximate p and then correcting bias by MC. Let us fix $q \ll d$ and define the active dimensions as $E_q = \{i_1, \ldots, i_q\} \subset \{1, \ldots, d\}$. Let us further denote with X^q the q dimensional vector $X^q = (X_{i_1}, \ldots, X_{i_q})$ and with X^{-q}

the $(d-q)$ dimensional vector $X^{-q} = (X_j)_{j \in E \setminus E_q}$. Then,

$$\begin{aligned} p = P(\max X > t) &= p_q + (1 - p_q) R_q, \\ p_q &= P(\max X^q > t), \\ R_q &= P(\max X^{-q} > t \mid \max X^q \leq t). \end{aligned} \tag{2}$$

The quantity p_q is always smaller or equal to p as $E_q \subset \{1, \ldots, d\}$. Selecting a non-degenerate vector X^q, we propose to estimate p_q with the QRSVN algorithm (Genz et al., 2012) which is efficient as we choose a number of active dimensions q much smaller than d. In Chevalier (2013), Chapter 6, the similar problem of approximating the non-exceedance probability of the maximum of a Gaussian random field (GRF) ξ based on a few well-selected points is presented. Each component of X stands for the value of ξ at one point of a given discretization of the field's domain. Active dimensions (i.e. the well-selected points) were chosen by numerically maximizing p_q, and the remainder was not accounted for. Our proposed method, instead, does not require a full optimization of the active dimensions as we exploit the decomposition in (2) to correct the error introduced by p_q. For this task, we propose two techniques to estimate the reminder R_q: a standard MC technique and a novel asymmetric nested Monte Carlo (anMC) algorithm. The anMC technique draws samples by taking into account the computational cost, resulting in a more efficient estimator.

In the remainder of the paper, we propose an unbiased estimator for p and we compute its variance in Section 2. In Section 3 we introduce the anMC algorithm in the more general setting of estimating expectations depending on two vectors with different simulation costs. It is then explicitly applied to efficiently estimate R_q. In Section 4 two numerical studies are presented. The first one studies the efficiency of the anMC algorithm compared to standard MC. The second one is a benchmark study where the efficiency of the proposed methods is compared with a selection of state-of-the-art techniques. Finally, in Section 5, we show an implementation of this method to compute conservative estimates of excursion sets for expensive to evaluate functions under non-necessarily Markovian Gaussian random field priors. More details on the choice of active dimensions are presented in Appendix A. All proofs are in Appendix B. Computer code for partially replicating the experiments presented here is attached in supplementary material.

2 The estimator properties

2.1 An unbiased estimator for p

Equation (2) gives us a decomposition that can be exploited to obtain an unbiased estimator for p. In the following proposition we define the estimator

and we compute its variance.

Proposition 1. *Consider $\widehat{p_q}$ and $\widehat{R_q}$, independent unbiased estimators of p_q and R_q respectively, then $\widehat{p} = \widehat{p_q} + (1 - \widehat{p_q})\widehat{R_q}$ is an unbiased estimator for p. Moreover its variance is*

$$\operatorname{var}(\widehat{p}) = (1 - R_q)^2 \operatorname{var}(\widehat{p_q}) + (1 - p_q)^2 \operatorname{var}(\widehat{R_q}) + \operatorname{var}(\widehat{p_q}) \operatorname{var}(\widehat{R_q}). \tag{3}$$

In what follows we consider different efficient computational strategies for $\widehat{p_q}$ and $\widehat{R_q}$.

2.2 Quasi Monte Carlo estimator for p_q

The quantity p_q can also be computed as

$$p_q = 1 - P\left(X^q \leq t_q\right),$$

where t_q denotes the q dimensional vector $(t, \ldots, t)$. The approximation of p_q thus requires only an evaluation of the c.d.f. of X^q. Since we assume that $q \ll d$, then the dimension is moderate and we propose to estimate p_q with the estimator $\widehat{p_q}$ that uses the method QRSVN introduced in Genz (1992), Hajivassiliou et al. (1996) and refined in Genz and Bretz (2009). This method computes a randomized quasi Monte Carlo integration of the normal density. In particular we consider here the implementation of QRSVN with the variable reordering described in Genz and Bretz (2009, Section 4.1.3). The estimate's error is approximated with the variance of the randomized integration. The quantity $\widehat{p_q}^{\mathrm{G}}$ obtained with this procedure is an unbiased estimator of p_q, see Genz and Bretz (2009).

The estimator $\widehat{p_q}^{\mathrm{G}}$ requires two choices: q, the number of active dimensions, and the dimensions themselves. The decomposition of p in Equation (2) leads to computational savings if we can approximate most of p with p_q for a small q. On the other hand a large number of active dimensions allows to intercept most of the probability mass in p. Here we adopt a heuristic approach to select both q and E_q sequentially by increasing the number of active dimensions until we meet an appropriate stopping condition. This approach, detailed in Algorithm 3, Appendix A, was chosen in the current implementation because it represents a good trade-off between speed and accuracy.

For a fixed q, the choice of E_q plays an important role in the approximation of p because it determines the error $\widehat{p_q} - p$, which is always negative. Selecting E_q such that $P(\max X^q > t)$ is numerically maximized, as in Chevalier (2013), optimally reduces the bias of $\widehat{p_q}$ as an estimator for p. Here we are not interested in a fully fledged optimization of this quantity as the residual bias is removed with the subsequent estimation of R_q, therefore,

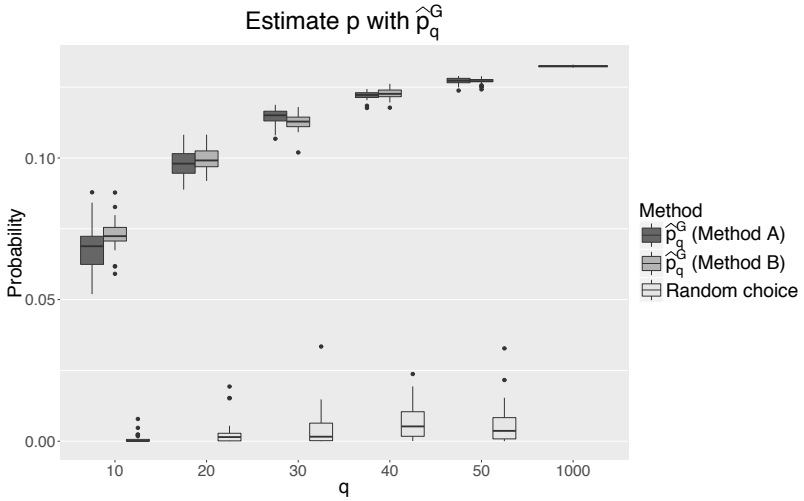


Figure 1: Distribution of $\widehat{p_q}^{\mathrm{G}}$ estimates obtained with different choices of active dimensions.

we exploit fast heuristics methods. The main tool used here is the excursion probability function:

$$p_t(i) = P(X_i > t) = \Phi\left(\frac{\mu_i - t}{\sqrt{\Sigma_{i,i}}}\right),$$

where Φ is the standard normal c.d.f. The function p_t is widely used in spatial statistics (see, e.g. Bolin and Lindgren, 2015) and Bayesian optimization (see, e.g. Kushner, 1964; Bect et al., 2012). In our setting it can be used to quickly identify the active dimensions. In fact this function takes high values at dimensions with high probability of exceeding the threshold and thus contribute the most to p. We propose the following methods.

Method A: sample q indices with probability given by p_t.

Method B: sample q indices with probability given by $p_t(1 - p_t)$.

These methods require only one evaluation of the normal c.d.f. at each element of the vector $\left(\frac{\mu_i - t}{\sqrt{\Sigma_{i,i}}}\right)_{i=1,\ldots,d}$, and are thus very fast. Both methods were already introduced for sequential evaluations of expensive to evaluate functions, see, e.g., Chevalier et al. (2014b).

Figure 1 shows a comparison of the estimates p_q obtained with different methods to select E_q. We consider 30 replications of an experiment where $\widehat{p_q}^{\mathrm{G}}$ is used to approximate p. The dimension of the vector X is $d = 1000$, the threshold is fixed at $t = 11$. The vector X is obtained from a discretization of a six dimensional GRF, defined on $[0, 1]^6$, over the first 1000 points of the Sobol' sequence (Bratley and Fox, 1988). The GRF has a tensor product Matérn ($\nu = 5/2$) covariance kernel. We generate a non

constant mean function $\mathfrak{m}$ by imposing the interpolation of a deterministic function at 70 points. The covariance kernel's hyperparameters are fixed as $\theta = [0.5, 0.5, 1, 1, 0.5, 0.5]^T$ and $\sigma^2 = 8$, see Rasmussen and Williams (2006), Chapter 4, for details on the parametrization. In this example, the two methods clearly outperform a random choice of active dimensions.

Methods A and B work well for selecting active dimensions when the mean vector μ and the covariance matrix diagonal are anisotropic. In such cases both methods select dimensions that are a good trade-off between high variance and mean close to t.

The choices of q and of the active dimensions influence the behaviour of the estimator for R_q. This aspect is discussed in more details in the next section.

2.3 Monte Carlo estimator for R_q

Debiasing $\widehat{p_q}^{\mathrm{G}}$ as an estimator of p can be done at the price of estimating

$$R_q = P\left(\max X^{-q} > t \mid \max X^q \leq t\right).$$

There is no closed formula for R_q, so it is approximated here via MC. Since X is Gaussian then so are X^q, X^{-q} and $X^{-q} \mid X^q = x^q$, for any deterministic vector $x^q \in \mathbb{R}^q$.

In order to estimate $R_q = P\left(\max X^{-q} > t \mid X_{i_1} \leq t, \ldots, X_{i_q} \leq t\right)$, we first generate n realizations $x_1^q, \ldots, x_n^q$ of X^q such that $X^q \leq t_q$. Second, we compute the mean and covariance matrix of X^{-q} conditional on each realization $x_l^q,\ l = 1, \ldots, n$ with the following formulas

$$\mu^{-q|x_l^q} = \mu^{-q} + \Sigma^{-q,q}(\Sigma^q)^{-1}(x_l^q - \mu^q), \qquad \Sigma^{-q|q} = \Sigma^{-q} - \Sigma^{-q,q}(\Sigma^q)^{-1}\Sigma^{q,-q}, \tag{4}$$

where μ^q, Σ^q and μ^{-q}, Σ^{-q} are the mean vector and covariance matrix of X^q and X^{-q} respectively, $\Sigma^{-q,q}$ is the cross-covariance between the dimensions $E \setminus E_q$ and E_q, $\Sigma^{q,-q}$ is the transpose of $\Sigma^{-q,q}$. Note that the conditional covariance $\Sigma^{-q|q}$ does not depend on the realization x_l^q, therefore it can be computed before the sampling procedure. Given the mean and covariance matrix conditional on each sample x_l^q, we can easily draw a realization $y_l^{-q|q}$ from $X^{-q} \mid X^q = x_l^q$. Once n couples $(x_l^q, y_l^{-q|q}),\ l = 1, \ldots, n$ are drawn from the respective distributions, an estimator for R_q is finally obtained as follows

$$\widehat{R_q}^{\mathrm{MC}} = \frac{1}{n}\sum_{l=1}^{n} \mathbf{1}_{\max y_l^{-q|q} > t}.$$

The realizations of X^q are obtained with a crude multivariate rejection sampling algorithm (Robert, 1995; Horrace, 2005). The cost of this step is driven by the acceptance probability of the sampler and it can be very high. The accepted samples satisfy the condition $X^q \leq t_q$ thus we have that the

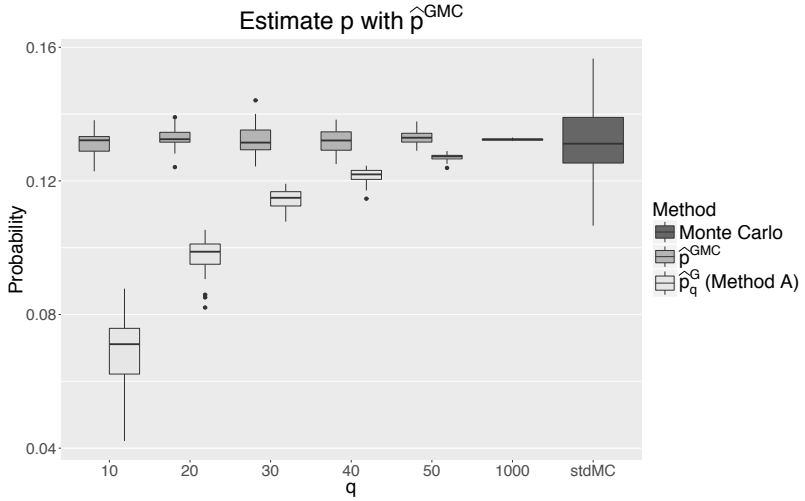


Figure 2: Estimate of p with $\widehat{p}^{\mathrm{GMC}}$ for different values of q. A full MC estimation of the same quantity is shown for comparison

acceptance probability is $P(X^q \leq t_q) = 1 - p_q$. This shows that the choice of q and of the active dimensions play an important role. If p_q is much smaller than p, then the rejection sampler will have a high acceptance probability, however the overall method will be less efficient as most of the probability is in the remainder. On the other hand, if q and the active dimensions are well chosen, the value of p_q could be very close to p. This will also lead to a slower rejection sampler as the acceptance probability would be small.

The second part of the procedure for $\widehat{R_q}^{\mathrm{MC}}$, drawing samples from the distribution of $X^{-q} \mid X^q = x_l^q$, is instead less dependent on q and generally less expensive than the fist step. The mean vector and covariance matrix computations requires only linear algebra operations as described in Equation (4) and realizations of $X^{-q} \mid X^q = x_l^q$ can be generated by sampling from a multivariate normal distribution.

The difference in computational cost between the first step and the second step of the MC procedure can be exploited to reduce the variance at a fixed computational cost. This idea is exploited by the asymmetric nested MC procedure presented in Section 3.

We denote with $\widehat{p}^{\mathrm{GMC}}$ the unbiased estimator of p defined as

$$\widehat{p}^{\mathrm{GMC}} = \widehat{p_q}^{\,\mathrm{G}} + (1 - \widehat{p_q}^{\,\mathrm{G}})\widehat{R_q}^{\mathrm{MC}},$$

where GMC denotes the use of Genz's method for p_q and MC for $\widehat{R_q}$.

Figure 2 shows the box plots of 30 replications of an experiment where p is approximated with $\widehat{p}^{\mathrm{GMC}}$. The set-up is the same as in Fig. 1. The core of the probability is approximated with $\widehat{p_q}^{\,\mathrm{G}}$ and the active dimensions are chosen with Method 1. The residual R_q is estimated with $\widehat{R_q}^{\mathrm{MC}}$. The re-

mainder allows to correct the bias of $\widehat{p_q}^{\mathrm{G}}$ even with a small number of active dimensions. As comparison the results of the same experiment with a full MC estimator for p are also shown. For all experiments and for each method the number of samples was chosen in order to have approximately the same computational cost. The estimator $\widehat{p}^{\mathrm{GMC}}$ exploits an almost exact method to estimate the largest part of the probability p, therefore the MC estimator $\widehat{R_q}^{\mathrm{MC}}$ has less variance than a full MC procedure for a fixed computational cost.

3 Estimation of the residual with asymmetric nested Monte Carlo

In section 2, R_q was estimated by $\widehat{R_q}^{\mathrm{MC}}$. There exists many methods to reduce the variance of such estimators, including antithetic variables (Hammersley and Morton, 1956), importance sampling (Kahn, 1950; Kahn and Marshall, 1953) or conditional Monte Carlo (Hammersley, 1956) among many others; see, e.g. Robert and Casella (2013, Chapter 4), for a broader overview. Here we focus on reducing the variance at a fixed computational cost, i.e. we are interested in increasing the estimator efficiency (Lemieux, 2009, Section 4.2). We propose a so-called asymmetric nested Monte Carlo (anMC) estimator for R_q that increases the efficiency with a parsimonious multiple use of conditioning data. In this section we develop some useful theoretical properties of anMC estimators.

The idea is to use an asymmetric sampling scheme that assigns the available computational resources by taking into account the actual cost of simulating each component. A similar asymmetric sampling scheme was introduced in the particular case of comparing the performance of stopping times for a real-valued stochastic process in discrete times in Dickmann and Schweizer (2016). Here we introduce this procedure in a general fashion and, in the next section, we detail it to $\widehat{R_q}^{\mathrm{MC}}$. For two measurable spaces $\mathcal{W}, \mathcal{Z}$, consider two random elements $W \in \mathcal{W}$ and $Z \in \mathcal{Z}$, defined on the same probability space and not independent. We are interested in estimating the quantity

$$G = \mathbb{E}\left[g(W, Z)\right], \tag{5}$$

where $g : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}$ is a measurable function, assumed integrable with respect to (W, Z)'s probability measure. Let us also assume that it is possible to draw realizations from the marginal distribution of W, Z and from the conditional distribution of $Z \mid W = w_i$, for each w_i sample of W. In the spirit of a Gibbs sampler, we can then obtain realizations (w_i, z_i), $i = 1, \ldots, n$ of (W, Z) by simulating w_i from the distribution of W and then z_i from the

conditional distribution $Z \mid W = w_i$, leading to:

$$\widehat{G} = \frac{1}{n} \sum_{i=1}^{n} g(w_i, z_i). \tag{6}$$

This MC estimator can actually be seen as the result of a two step nested MC procedure where, for each realization w_i, one inner sample z_i is drawn from $Z \mid W = w_i$. Note that the estimator $\widehat{R_q}^{\mathrm{MC}}$ used in Section 2 is a particular case of Equation (6) with $W = X^q \mid X^q \leq t_q$, $Z = X^{-q}$ and $g(x, y) = \mathbf{1}_{\max y > t}$. As noted in Section 2, drawing realizations of $X^q \mid X^q \leq t_q$ has a higher computational cost than simulating X^{-q} because rejection sampling is required in the first case. More generally, let us denote with $C_W(n)$ the cost of n realizations of W and with $C_{Z|W}(m; w_i)$ the cost of drawing m conditional simulations from $Z \mid W = w_i$. If $C_W(1)$ is much higher than $C_{Z|W}(1; w_i)$ then sampling several conditional realizations for a given w_i might bring computational savings.

In the proposed asymmetric sampling scheme for each realization w_i we sample m realizations $z_{i,1}, \ldots, z_{i,m}$ from $Z \mid W = w_i$. Assume that we use this sampling scheme for the couples $(w_i, z_{i,j})$, $i = 1, \ldots, n,\ j = 1, \ldots, m$, then an estimator for G is

$$\widetilde{G} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} g(w_i, z_{i,j}). \tag{7}$$

For a fixed number of samples, the estimator $\widetilde{G}$ may have a higher variance than $\widehat{G}$ due to the dependency between pairs sharing the same replicate of W. However, in many cases, the estimator $\widetilde{G}$ may be relevant to reduce the variance at a fixed computational time. In fact, let us fix the computational budget instead of the number of samples. If $C_{Z|W}(1; w_i) < C_W(1)$, then anMC may lead to an overall variance reduction thanks to an increased number of simulated pairs. In the remainder of the section, we show that, in the case of an affine cost functions, there exists an optimal number of inner simulations m such that $\mathrm{var}(\widetilde{G}) < \mathrm{var}(\widehat{G})$. Assume

$$\begin{aligned} C_W(n) &= c_0 + cn \text{ and, for each sample } w_i \\ C_{Z|W}(m; w_i) &= C_{Z|W}(m) = \alpha + \beta m, \end{aligned}$$

with $c_0, c, \alpha, \beta \in \mathbb{R}_+$ dependent on the simulators of W and $Z \mid W$. The second equation entails that the cost of conditional simulations does not depend on the conditioning value. If $W = X^q \mid X^q \leq t_q$, $Z = X^{-q}$ as in Section 2, then $Z \mid W$ is Gaussian with mean and covariance matrix described in (4). In this case, the cost for sampling $Z \mid W$ is affine, with α describing preliminary computations and β random number generation and algebraic operations. Denote with $W_1, \ldots, W_n$ replications of W. For

each W_i we consider the conditional distribution $Z \mid W_i$ and m replications $Z_{1,i}, \dots, Z_{m,i}$. Under these assumption the total simulation budget is

$$C_{\text{tot}}(n, m) = c_0 + n(c + \alpha + \beta m).$$

If the total budget is fixed, $C_{\text{tot}}(n, m) = C_{\text{fix}} \in \mathbb{R}_+$, then the number of replications of W as a function of m is

$$N_{C_{\text{fix}}}(m) = \frac{C_{\text{fix}} - c_0}{c + \alpha + \beta m}.$$

The following proposition shows a decomposition of $\text{var}(\widetilde{G})$ that is useful to find the optimal number of simulations m^* under a fixed simulation budget $C_{\text{tot}}(n, m) = C_{\text{fix}}$.

Proposition 2. *Consider n independent copies $W_1, \dots, W_n$ of W and, for each W_i, m copies $Z_{i,j} = Z_j \mid W_i$ $j = 1, \dots, m$, independent conditionally on W_i. Then,*

$$\text{var}(\widetilde{G}) = \frac{1}{n} \text{var}(g(W_1, Z_{1,1})) - \frac{m-1}{nm} \mathbb{E}\big[\text{var}(g(W_1, Z_{1,1}) \mid W_1) \big]. \quad (8)$$

Corollary 1. *Under the same assumptions, $\widetilde{G}$ has minimal variance when*

$$m = \widetilde{m} = \sqrt{\frac{(\alpha + c)B}{\beta(A - B)}},$$

where $A = \text{var}(g(W_1, Z_{1,1}))$ and $B = \mathbb{E}\big[\text{var}(g(W_1, Z_{1,1}) \mid W_1) \big]$. Moreover denote with $\varepsilon = \widetilde{m} - \lfloor \widetilde{m} \rfloor$, then the optimal integer is $m^* = \lfloor \widetilde{m} \rfloor$ if

$$\varepsilon < \frac{(2\widetilde{m} + 1) - \sqrt{4(\widetilde{m})^2 + 1}}{2} \quad (9)$$

or $m^* = \lceil \widetilde{m} \rceil$ otherwise.

Proposition 3. *Under the same assumptions, if $m^* > \frac{2(\alpha+c)B}{(c+\alpha)B + \beta(A-B)}$ then $\text{var}(\widetilde{G}) = \text{var}(\widehat{G})\,[1 - \eta]$, where $\eta \in (0, 1)$.*

3.1 Algorithmic considerations

In order to compute m^*, we need the quantities $A = \text{var}(g(W_1, Z_{1,1}))$ and $B = \mathbb{E}\big[\text{var}(g(W_1, Z_{1,1}) \mid W_1) \big]$ and the constants c_0, c, α and β. A and B depend on the specific problem at hand and are usually not known in advance. Part of the total computational budget is then needed to estimate A and B. This preliminary phase is also used to estimate the system dependent constants c and β. Algorithm 1 reports the pseudo-code for anMC.

Algorithm 1: Asymmetric nested Monte Carlo.

Input : $\mu_W, \mu_Z, \Sigma_W, \Sigma_Z, \Sigma_{WZ}, g, C_{\text{tot}}$

Output: $\widetilde{G}$

Part 0: estimate c_0, c, β, α ;

initialize compute the conditional covariance $\Sigma_{Z|W}$ and initialize n_0, m_0;

Part 1: **for** $i \leftarrow 1$ **to** n_0 **do**

estimate A, B

simulate w_i from the distribution of W and compute $\mu_{Z|W=w_i}$;

draw m_0 simulations $z_{i,1}, \ldots, z_{i,m_0}$ from the conditional distribution $Z \mid W = w_i$;

estimate $\mathbb{E}[g(W,Z) \mid W = w_i]$ with $\tilde{E}_i = \frac{1}{m_0} \sum_{j=1}^{m_0} g(w_i, z_{i,j})$;

estimate $\operatorname{var}(g(W,Z) \mid W = w_i)$ with $\tilde{V}_i = \frac{1}{m_0 - 1} \sum_{j=1}^{m_0} (g(w_i, z_{i,j}) - \tilde{E}_i)^2$;

end

compute $\widetilde{m} = \sqrt{\frac{(\alpha+c)\frac{1}{n_0}\sum_{i=1}^{n_0} \tilde{V}_i}{\beta \frac{1}{n_0 - 1} \sum_{i=1}^{n_0} (\tilde{E}_i - \frac{1}{n_0}\sum_{i=1}^{n_0} \tilde{E}_i)^2}}$, m^* as in Corollary 1 and $n^* = N_{C_{\text{fix}}}(m^*)$;

Part 2: **for** $i \leftarrow 1$ **to** n^* **do**

compute $\widetilde{G}$

if $i \leq n_0$ **then**

for $j \leftarrow 1$ **to** m^* **do**

if $j \leq m_0$ **then**

use previously calculated $\widetilde{E}_i$ and $\widetilde{V}_i$;

else

simulate $z_{i,j}$ from the distribution $Z \mid W = w_i$;

compute $\widetilde{E}_i = \frac{1}{m^*} \sum_{j=1}^{m^*} g(w_i, z_{i,j})$;

end

end

else

simulate w_i from the distribution of W and compute $\mu_{Z|W=w_i}$;

for $j \leftarrow 1$ **to** m^* **do**

simulate $z_{i,j}$ from the conditional distribution $Z \mid W = w_i$;

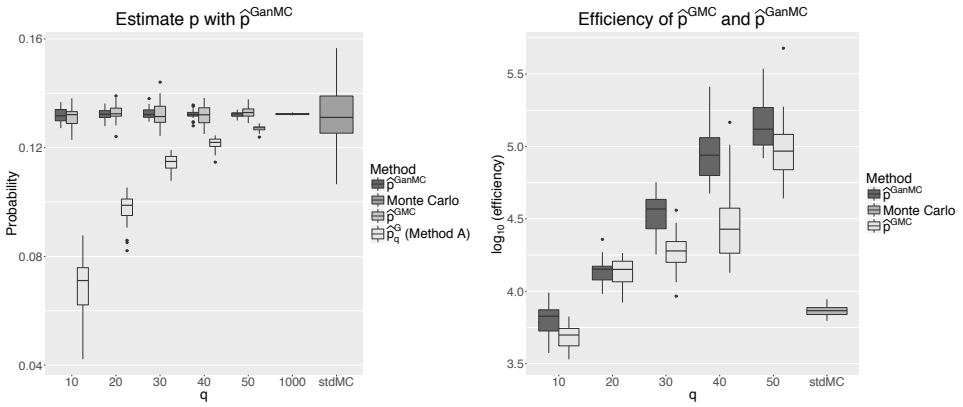
end

compute $\widetilde{E}_i = \frac{1}{m^*} \sum_{j=1}^{m^*} g(w_i, z_{i,j})$;

end

end

estimate $\mathbb{E}[g(W,Z)]$ with $\widetilde{G} = \frac{1}{n^*} \sum_{i=1}^{n^*} \tilde{E}_i$;



(a) Probability values. (b) Efficiency. Values in logarithmic scale.

Figure 3: Comparison of results with $\widehat{p}_q^{\,\mathrm{G}}$, $\widehat{p}^{\mathrm{GMC}}$, $\widehat{p}^{\mathrm{GanMC}}$ and standard MC on 30 replications of the example introduced in Fig. 1.

3.2 Estimate p with $\widehat{p}^{\mathbf{GanMC}}$

The anMC algorithm can be used to reduce the variance compared to R_q's MC estimate proposed in Section 2.3. In fact, let us consider $W = X^q \mid X^q \leq t_q$ and $Z = X^{-q}$. We have that W is expensive to simulate as it requires rejection sampling while, for a given sample w_i, $Z \mid W = w_i$ is Gaussian with mean and covariance matrix described in Equation (4). It is generally much cheaper to obtain samples from $Z \mid W = w_i$ than from W. Moreover, as noted earlier, R_q can be written in the form of Equation (5) with $g(x, y) = \mathbf{1}_{\max y > t}$. By following Algorithm 1 we calculate m^*, sample n^* realizations $w_1, \ldots, w_{n^*}$ of W and for each realization w_i obtain m^* samples $z_{i,1}, \ldots, z_{i,m^*}$ of $Z \mid W = w_i$. We estimate R_q via

$$\widehat{R_q}^{\mathrm{anMC}} = \frac{1}{n^* m^*} \sum_{i=1}^{n^*} \sum_{j=1}^{m^*} \mathbf{1}_{\max z_{i,j} > t}.$$

Finally plugging in $\widehat{R_q}^{\mathrm{anMC}}$ and $\widehat{p}_q^{\,\mathrm{G}}$ in Equation (2), we obtain

$$\widehat{p}^{\mathrm{GanMC}} = \widehat{p}_q^{\,\mathrm{G}} + (1 - \widehat{p}_q^{\,\mathrm{G}}) \widehat{R_q}^{\mathrm{anMC}}.$$

Figure 3a shows a comparison of results using 30 replications of the experiment presented in Section 2.3. Results obtained with a MC estimator are shown for comparison.

While the simulations of all experiments were obtained under the constraint of a fixed computational cost, the actual time to obtain the simulations was not exactly the same. In order to be able compare the methods in more general settings we further rely on the notion of efficiency. For an

estimator $\widehat{p}$, we define the efficiency (Lemieux, 2009, Section 4.2) as

$$\mathrm{Eff}[\widehat{p}] = \frac{1}{\mathrm{var}(\widehat{p})\,\mathrm{time}[\widehat{p}]},$$

where $\mathrm{time}[\widehat{p}]$ denotes the computational time of the estimator $\widehat{p}$.

Figure 3b shows a comparison of the efficiency of $\widehat{p}^{\mathrm{GMC}}$ and $\widehat{p}^{\mathrm{GanMC}}$ with a full Monte Carlo estimator. With as few as $q = 50$ active dimensions we obtain an increase in efficiency of around 10 times on average over the 30 replications of the experiment with the estimator $\widehat{p}^{\mathrm{GMC}}$. The estimator $\widehat{p}^{\mathrm{GanMC}}$ shows a higher median efficiency than the others for all $q \geq 20$.

4 Numerical studies

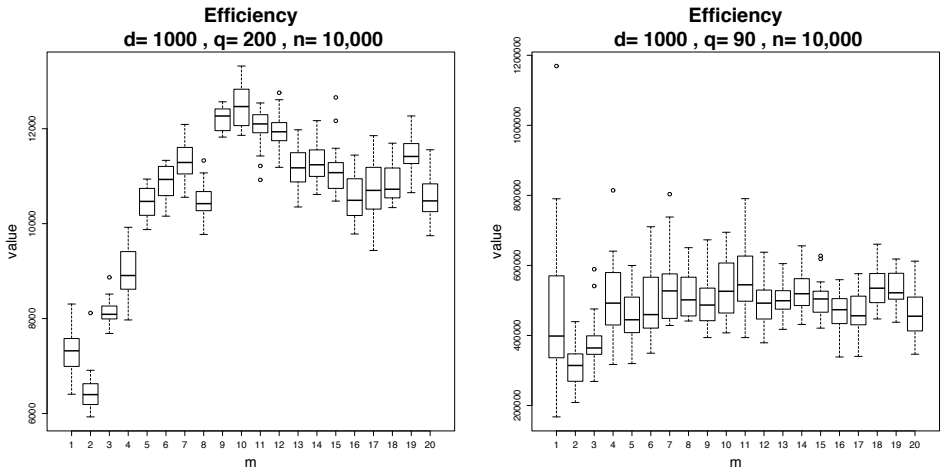
4.1 Choice of the number of inner samples

In this section we study the efficiency of the anMC method compared with a standard MC method for different choices of m. Here we do not select the optimal m^* defined in Corollary 1, but we study the efficiency as a function of m. In many practical situations even if part 1 of Algorithm 1 does not render the optimal m^* the anMC algorithm is still more efficient than a standard MC if the chosen m is close to m^*.

We consider a similar setup to the experiment presented in Section 2.2. Here we start from a GRF with tensor product Matérn ($\nu = 5/2$) and a non constant mean function $\mathfrak{m}$ different from the example in Section 2.2, initialized as conditional mean on 60 randomly generated values at a fixed design on $[0,1]^6$. The hyperparameters are fixed as $\theta = [0.5, 0.5, 1, 1, 0.5, 0.5]^T$ and $\sigma^2 = 8$. The GRF is then discretized over the first $d = 1000$ points of the Sobol sequence to obtain the vector X. We are interested in $1 - p = P(X < t)$, with $t = 5$. We proceed by estimating p with $\widehat{p}^{\mathrm{GMC}}$ and $\widehat{p}^{\mathrm{GanMC}}$ for different choices of m to compare their efficiency. The initial part p_q is computed once with estimator $\widehat{p_q}^{\mathrm{G}}$ with q and the active dimensions chosen with Algorithm 3, Method B. The number of outer simulations in the anMC algorithm is kept fixed to $n = 10,000$ and we only vary m. For each m, the anMC estimation is replicated 20 times.

The median estimated value for p is $\widehat{p} = 0.9644$. Most of the probability is estimated with p_q, in fact $\widehat{p_q}^{\mathrm{G}} = 0.9636$. Figure 4a shows $\mathrm{Eff}[\widehat{p}]$ computed with the overall variance of $\widehat{p}$. A choice of $m = 10$ leads to a median increase in efficiency of 73% compared to the MC case. In this example, both the probability to be estimated and p_q are close to 1, thus the acceptance probability for $\widehat{R_q}$ is low. In this situation the anMC method is able to exploit the difference in computational costs to provide a more efficient estimator for $\widehat{R_q}$.

In order to study the effect of the acceptance probability on the method's efficiency we change the threshold in the previous example to $t = 7.5$ by



(a) High probability state, $t = 5$, $\widehat{p}^{\text{GanMC}} = 0.9644$.

(b) Low probability state, $t = 7.5$, $\widehat{p}^{\text{GanMC}} = 0.1178$.

Figure 4: Efficiency of $\widehat{p}^{\text{GanMC}}$ estimator versus the number of inner simulations m. For each m the experiment is reproduced 30 times.

keeping the remaining parameters fixed. The value of p is smaller, $\widehat{p} = 0.1178$. The number of active dimensions q, chosen with Algorithm 3, is smaller ($q = 90$) as the probability mass is smaller. The value of p_q ($\widehat{p_q}^{\text{G}} = 0.1172$) is much smaller than in the previous case and this leads to a higher acceptance probability for $\widehat{R_q}$. Figure 4b shows efficiency of the method as a function of m. Here the anMC method does not bring significant gains over the MC method as the the ratio between the cost of rejection sampling and the conditional simulations in $\widehat{R_q}$ is close to one. The estimated m^* is equal to 1.91, thus it is smaller than the minimum threshold of Proposition 3 that guarantees a more efficient anMC algorithm.

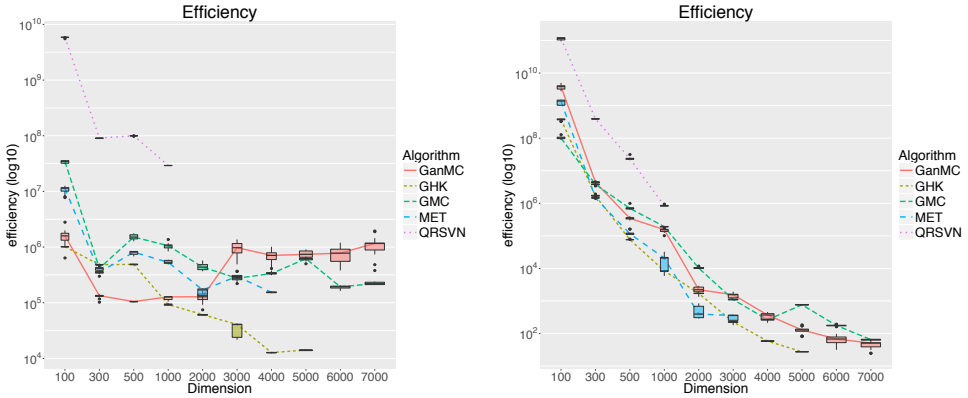
4.2 Comparison with state of the art

In this section we compare the GMC and GanMC methods, as implemented in the R package `ConservativeEstimates`, with available state-of-the-art algorithms to estimate $\pi(t)$. In particular, we compare this implementation with:

QRSVN an implementation of Genz method (Genz and Bretz, 2009) in the R package `mvtnorm`, function `pmvnorm`;

GHK an implementation of GHK method (Geweke, 1991; Hajivassiliou and McFadden, 1998) in the R package `bayesm`, function `ghkvec`;

MET R implementation of the minimax-exponentially-tilted (MET) method (Botev, 2017) in the package `TruncatedNormal`, function `mvNcdf`;.



(a) Low $\pi(t)$ state, $t = 5$. The median estimated value for $p = 1 - \pi(t)$ ranges from 0.33845 to 0.99876.

(b) High $\pi(t)$ state, $t = 7.5$. The median estimated value for $p = 1 - \pi(t)$ ranges from 0.00315 to 0.32564.

Figure 5: Efficiency of the probability estimator versus the dimension d. For each d the experiment is reproduced 15 times. Values in logarithmic scale.

We consider the example introduced in Section 4.1 and we increase the dimension of the problem d by considering finer discretizations of the underlying GRF. For example, the vector X of dimension $d = 100$ is obtained from the GRF discretized on the first 100 points of the 6-dimensional Sobol' sequence. As the dimension d increases the probability $\pi(t)$ changes, thus providing different setups. Each experiment is replicated 15 times.

Figure 5a presents a comparison of the estimator's efficiency for the problem of computing $\pi(t)$, with $t = 5$. This is a low probability setup, as the range of $\pi(t)$ varies between 0.66155 for $d = 100$ and 0.00124 for $d = 7000$. The most efficient algorithm is the QRSVN Genz method, however this implementation does not scale to dimensions higher than 1000. The GMC algorithm is the second most efficient in all dimensions except $d = 2000$ where it is the most efficient. The GanMC algorithm is instead the most efficient when d is greater than 2000. This effect is explained by the efficiency gains brought by $\widehat{R}_q^{\text{anMC}}$ when the rejection sampler is expensive. If $d > 2000$, the probability $P(X^q \leq t_q)$ is always smaller than 0.01, thus the rejection sampler becomes much more expensive than the conditional sampler in the estimation of the remainder $\widehat{R}_q$. Algorithms GHK and MET allowed estimates until dimension $d = 5000$ and $d = 4000$ respectively before running in memory overflows. The GanMC algorithm is 45 times more efficient than the GHK algorithm for $d = 5000$ and 3.8 times more efficient than MET for $d = 4000$. It is also 5 times more efficient than GMC for $d = 7000$.

Figure 5b compares the estimators' efficiency for the computation of $\pi(t)$ with $t = 7.5$. As partially observed in the previous section this is a

high probability setup as the median estimate of $\pi(t)$ ranges from 0.99685, for $d = 100$, to 0.67436, for $d = 7000$. Also in this case the QRSVN is the most efficient algorithm in low dimensions. The GMC and the GanMC algorithms however are the most efficient for all dimensions higher than 2000. The GanMC algorithm is 4 times more efficient than the MET for $d = 3000$ and 5 times more efficient than GHK for $d = 5000$. In this setup the computational cost of the rejection sampler in $\widehat{R_q}^{\text{anMC}}$ is not much higher than the conditional sampler. In fact, the acceptance probability of the rejection sampler is always higher than 0.6. In most replications this leads to a choice of m^* smaller than 2 and thus GanMC is slower than GMC because of Part 1 in Algorithm 1 while achieving the same variance. This is the main reason why the GMC algorithm proves to be more efficient for most dimensions, in fact for $d = 5000$ it is 5.9 times more efficient than GanMC and for $d = 7000$ the ratio is 1.3. All computations were carried on the cluster of the University of Bern on machines with Intel Xeon CPU 2.40GHz and 16 GB RAM.

5 Application: efficient computation of conservative estimates

We show here that anMC is key in conservative excursion set estimation relying on Gaussian field models. We consider an expensive to evaluate system described by a continuous function $f : D \subset \mathbb{R}^\ell \to \mathbb{R}, \ell \geq 1$, where D is a compact domain, and we focus on estimating, for some fixed threshold $t \in \mathbb{R}$, the set

$$\Gamma^* = \{x \in D : f(x) \leq t\}.$$

Such problems arise in many applications such as reliability engineering (see, e.g., Picheny et al. (2013), Chevalier et al. (2014a)) climatological studies (Bolin and Lindgren, 2015; French and Sain, 2013) or in natural sciences (Bayarri et al., 2009). Often f is seen as expensive to evaluate black-box (Sacks et al., 1989) and can only be evaluated with computer simulations. We assume here that f was only evaluated at points $\chi_k = \{x_1, \ldots, x_k\} \subset D$ and the associated responses are denoted with $f(\chi_k) = (f(x_1), \ldots, f(x_k)) \in \mathbb{R}^k$ and we are interested in giving an estimate of Γ^* starting from these k evaluations.

In a Bayesian framework we consider f as a realization of a GRF $(\xi_x)_{x \in D}$ with prior mean function $\mathfrak{m}$ and covariance kernel $\mathfrak{K}$. A prior distribution of the excursion set is hence obtained by thresholding ξ, thus obtaining the following random closed set

$$\Gamma = \{x \in D : \xi_x \leq t\}.$$

Denoting with ξ_{χ_k} the random vector $(\xi_{x_1}, \ldots, \xi_{x_k})$, we can then condition ξ on the observations $f(\chi_k)$ and obtain a posterior distribution for the field

$\xi_x \mid \xi_{\chi_k} = f(\chi_k)$. This gives rise to a posterior distribution for Γ. Different definitions of random closed set expectation (Molchanov (2005), Chapter 2) can be used to summarize this posterior distribution and to provide estimates for Γ^*. In Chevalier et al. (2013), for example, the Vorob'ev expectation was introduced in this setting. Let us briefly recall this definition. We denote with $p_{\Gamma,k} : D \to [0,1]$ the coverage function of the posterior set $\Gamma \mid \xi_{\chi_k} = f(\chi_k)$, defined as

$$p_{\Gamma,k}(x) = P_k(x \in \Gamma),\ x \in D,$$

where $P_k(\cdot) = P(\cdot \mid \xi_{\chi_k} = f(\chi_k))$. This function associates to each point in D its probability of being inside the posterior excursion set. The function $p_{\Gamma,k}$ gives rise to a family of excursion set estimates: for each $\rho \in [0,1]$ we can define the posterior ρ-level Vorob'ev quantile of Γ

$$Q_\rho = \{x \in D : p_{\Gamma,k}(x) \geq \rho\}.$$

The Vorob'ev expectation of Γ (Molchanov, 2005) is the quantile Q_{ρ_V} that satisfies $|Q_\rho| \leq \mathbb{E}_k[|\Gamma|] \leq |Q_{\rho_V}|$ for all $\rho \geq \rho_V$, where $|A|$ denotes the volume of a set $A \subset \mathbb{R}^l$. The set Q_{ρ_V} consists of the points that have high enough marginal probability of being inside the excursion set. In some applications, however, it is important to provide confidence statements on the whole set estimate. Conservative estimates introduced in Bolin and Lindgren (2015) for Gaussian Markov random fields address this issue. A conservative estimate of Γ^* is

$$C_{\Gamma,k} = \arg\max_{C \subset D} \{|C| : P_k(C \subset \{\xi_x \leq t\}) \geq \alpha\}, \tag{10}$$

where $|C|$ denotes the volume of C.

The object in Equation (10), however, leads to major computational issues. First of all we need to select a family of sets to use for the optimization procedure in Equation (10). Here we follow Bolin and Lindgren (2015) and select the Vorob'ev quantiles as family of sets. This family has the advantage that it is parametrized by one real number ρ and thus it renders the optimization straightforward. Algorithm 2 details the optimization procedure.

Second, for each candidate Q we need to evaluate $P_{\text{next}} = P_k(Q \subset \{\xi_x \leq t\})$, the probability that Q is inside the excursion. In fact, this quantity is a high dimensional orthant probability. For a Vorob'ev quantile $Q_{\rho\prime}$, discretized over the points $c_1, \ldots, c_r$,

$$P_k(Q_{\rho'} \subset \{\xi_x \leq t\}) = P_k(\xi_{c_1} \leq t, \ldots, \xi_{c_r} \leq t) = 1 - P_k(\max_{i=1,\ldots,r} \xi_{c_i} > t).$$

Thus we use the estimator $\widehat{p}^{\text{GanMC}}$ to approximate $1 - P_k(Q_{\rho'} \subset \{\xi_x \leq t\})$. The use of anMC allows resolutions for the discretized Vorob'ev quantiles that seem out of reach otherwise.

Algorithm 2: Conservative estimates algorithm.

Input : $\mathfrak{m}_k, \mathfrak{K}_k$, conditional mean and covariance of $\xi \mid \xi_{\chi_k} = f(\chi_k)$, and G, fine discretization design;

Output: Conservative estimate for Γ^* at level α.

Part 0: sort the points in G in decreasing order of $p_{\Gamma,k}$, with indices $G_s = \{i_1, \dots i_m\}$;

compute i_B, i_T find the highest index i_T such that $\prod_{j=1}^{T} p_{\Gamma,k}(G_s)[i_j] \geq \alpha$;
find the highest index i_B such that $p_{\Gamma,k}(G_s)[i_B] \geq \alpha$;
evaluate mean and covariance matrix $\mathfrak{m}_k(i_B)$ and Σ_{i_B,i_B};

Part 1: initialize $i_L = i_T$, $i_R = i_B$;

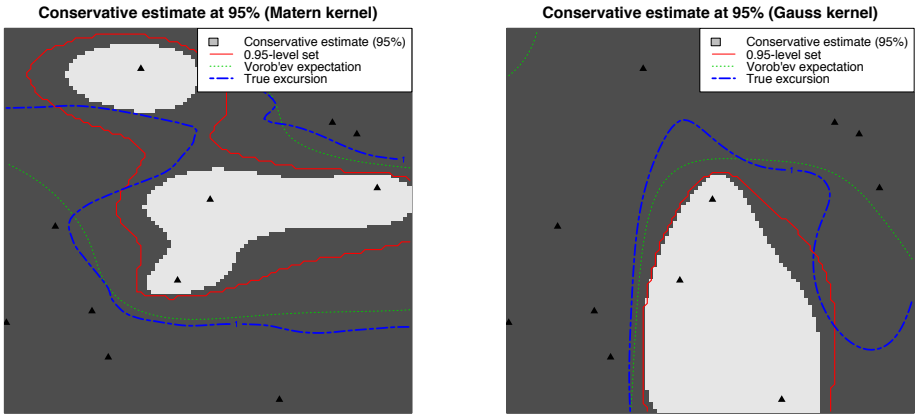
Initialize dichotomy estimate $P_L = P_k(Q_{\rho_{i_L}} \subset \{\xi_x \leq t\})$, $P_R = P_k(Q_{\rho_{i_R}} \subset \{\xi_x \leq t\})$ with GanMC ;

Part 2: **while** $P_R < \alpha$ *and* $(i_R - i_L) \geq 2$ **do**

optimization
- next evaluation $i_{\text{next}} = \frac{i_L + i_R}{2}$;
- estimate $P_{\text{next}} = P_k(Q_{\rho_{i_{\text{next}}}} \subset \{\xi_x \leq t\})$ with GanMC;
- **if** $P_{next} \geq \alpha$ **then**
  - $i_L = i_{\text{next}}$, $i_R = i_R$;
- **else**
  - $i_L = i_L$, $i_R = i_{\text{next}}$;
- **end**

end

We apply Algorithm 2 to a two dimensional artificial test case. We consider as function f a realization of a GRF $(\xi_x)_{x \in D}$, where $D \subset \mathbb{R}^2$ is the unit square. We consider two parametrizations for the prior covariance kernel: a tensor product Matérn covariance kernel with $\nu = 5/2$, variance $\sigma^2 = 0.5$ and range parameters $\theta = [0.4, 0.2]$ and a Gaussian covariance kernel with variance $\sigma^2 = 0.5$ and range parameters $\theta = [0.2, 0.4]$. In both cases we assume a prior constant mean function. We are interested in the set Γ^* with $t = 1$. For both cases we consider $k = 15$ evaluations of f at the same points chosen by Latin hypercube sampling. Figures 6a and 6b show the conservative estimate at level 95% compared with the true excursion, the Vorob'ev expectation and the 0.95-quantile for the Matérn and the Gaussian kernel. The 0.95-quantile does not guarantee that the estimate is included in the true excursion with probability 0.95 in both examples. The conservative estimates instead are guaranteed to be inside the true excursion with probability $\alpha = 0.95$. They correspond to Vorob'ev quantiles at levels 0.998 (Matérn) and 0.993 (Gaussian). The conservative estimates were obtained with a 100×100 discretization of the unit square. Such high resolution grids lead to very high dimensional probability calculations. In fact, the dichotomy algorithm required 11 computations of the probability $1 - P_k(Q_{\rho'} \subset \{\xi_x \leq t\})$ for each case. The discretization's size for Q_ρ varied



(a) Realization obtained with a Matérn kernel.

(b) Realization obtained with Gaussian kernel.

Figure 6: Conservative estimates at 95% (white region) for the excursion below $t = 1$. Both models are based on 15 evaluations of the function (black triangles). The true excursion level is plotted in blue, the Vorob'ev expectation in green and the 0.95-level set in red.

between 1213 and 3201 points in the Matérn kernel case and between 1692 and 2462 points in the Gaussian case. Such high dimensional probabilities cannot be computed with the current implementation of the algorithm by Genz, however they could be computed with other Monte Carlo methods at higher computational costs. Instead, with the proposed method, the total computational time on a laptop with Intel Core i7 1.7GHz CPU and 8GB of RAM was equal to 365 and 390 seconds respectively for Matérn and Gaussian kernel.

6 Discussion

In this paper we introduced a new method to approximate high dimensional orthant Gaussian probabilities based on a decomposion of the probability in a low dimensional part p_q and a remainder R_q. The number of active dimensions q and the dimensions themselves are chosen with two heuristic algorithms which provide good results in case of dense covariance matrix with anisotropic diagonal and anisotropic mean vector. An alternative proposal is choosing the first q dimensions ordered according to the inverse Genz variable reordering proposed in Genz and Bretz (2009, Section 4.1.3). While similar to the heuristics proposed here, this method is not efficient in high dimensions as it requires a full Cholesky decomposition of the covariance matrix. The remainder R_q is instead estimated with two methods:

standard Monte Carlo and asymmetric nested Monte Carlo (anMC). Both methods showed higher efficiency than other state-of-the-art methods for dimensions higher than 1000.

The anMC method proved to be very efficient, in the numerical studies presented, if the orthant probability of interest has low to medium values. This method however relies on an initial step where several constants and probabilistic quantities are empirically estimated to choose the optimal m^*, the number of inner samples. In particular the cost parameters c, β, the slopes of the linear costs, might be hard to estimate if the constants c_0, α are comparatively large. In this case Algorithm 1 might not choose the optimal m^*. However, a numerical study of the algorithm behaviour for different choices of m showed that, on the considered examples, even if the chosen m is not optimal but it is close to optimal, the efficiency gain is very close to the optimal efficiency gain. The estimator $\widehat{p}^{\text{GanMC}}$ efficiency is mainly driven by the acceptance probability of the rejection sampler in $\widehat{R_q}^{\text{anMC}}$, which depends on $\widehat{p_q}$. This highlights the existence of a trade-off between $\widehat{p_q}^{\text{G}}$ and $\widehat{R_q}$. If the choice of q and active dimensions is not optimal, then the acceptance probability of the rejection sampler becomes larger, making the estimation of $\widehat{R_q}$ easier. An estimator $\widehat{p_q}$ closer to p makes the quantity $\widehat{R_q}$ harder to estimate. However, in this case, $\widehat{R_q}^{\text{anMC}}$ becomes more efficient than $\widehat{R_q}^{\text{MC}}$ as the ratio between the computational costs becomes more favourable.

The estimator $\widehat{p}^{\text{GanMC}}$ made possible the computation of conservative estimates of excursion sets with general GRF priors. The R implementation of the algorithm is contained in the package `ConservativeEstimates` currently available on GitHub.

A Choice of active dimensions

The estimator $\widehat{p_q}^{\text{G}}$, introduced in Section 2.2, requires the choice of q, the number of active dimensions and the choice of the dimensions themselves. Algorithm 3 describes the heuristic procedure implemented in `ConservativeEstimates` to select q and obtain the active dimensions. Here we select q by sequentially increasing the number of active dimensions until the relative change of $\widehat{p_q}^{\text{G}}$ is less than the estimate's error.

The constant $\gamma > 0$ is chosen equal to 1 in our implementation. Moreover the algorithm stops if $q_k > 300$ to avoid using Genz's algorithm in high dimensions.

A.1 Add spatial information

If the random vector X comes from a GRF discretized over a set of points $E_{\text{spat}} = \{e_1, \dots, e_d\} \subset \mathbb{R}^l$, then we can exploit this information to choose

Algorithm 3: Select q, active dimensions and compute $\widehat{p_q}^{\mathrm{G}}$.

Input : q_0, small initial q, e.g. $q_0 = d^{1/3}$, and q_{step} the increment of q, $\gamma > 0$
Output: q, $\widehat{p}_q^{\mathrm{G}}$

Compute $\widehat{p}_{q_0}^{\mathrm{G}}$ and save $\mathrm{err}(\widehat{p}_{q_0}^{\mathrm{G}}) := 3\sqrt{\mathrm{var}(\widehat{p}_{q_0}^{\mathrm{G}})}$;
initialize $k = 0$;
repeat
  increment $k = k + 1$;
  $q_k := q_0 + kq_{\text{step}}$;
  choose q_k active dimensions, compute $\widehat{p}_{q_k}^{G}$ and $\mathrm{err}(\widehat{p}_{q_k}^{\mathrm{G}})$;
  compute $\Delta(\widehat{p}_{q_k}^{\mathrm{G}}) = \frac{\left|\widehat{p}_{q_k}^{\mathrm{G}} - \widehat{p}_{q_{k-1}}^{\mathrm{G}}\right|}{1+\widehat{p}_{q_k}^{\mathrm{G}}}$;
until $\Delta(\widehat{p}_{q_k}^{G}) < \gamma\,\mathrm{err}(\widehat{p}_{q_k}^{G})$ *or* $q_k > 300$;
$q = q_k$ and $\widehat{p_q}^{\mathrm{G}} = \widehat{p}_{q_k}^{\mathrm{G}}$;

E_q. Let us consider the sequence of vectors $(\boldsymbol{\delta}_j)_{j=1,\dots,q}$, defined for each j as

$$\boldsymbol{\delta}_j = \prod_{k=1}^{j} \mathrm{dist}(e_{i_k}, E_{\text{spat}}) \quad (j = 1, \dots, q)$$

where $\mathrm{dist}(e_{i_k}, E_{\text{spat}})$ denotes the d-dimensional vector of Euclidean distances between e_{i_k} and each point in E_{spat} and $\{e_{i_1}, \dots, e_{i_q}\}$ are the points corresponding to the selected active dimensions E_q. We then adjust Methods A, B by sampling the jth active dimension with probabilities given by the component-wise products $p_t \frac{\boldsymbol{\delta}_j}{\|\boldsymbol{\delta}_j\|}$ and $p_t(1-p_t)\frac{\boldsymbol{\delta}_j}{\|\boldsymbol{\delta}_j\|}$ respectively.

B Proofs

Proof of Proposition 1

Proof. We have that $\mathbb{E}[\widehat{p_q}] = p_q$ and $\mathbb{E}[\widehat{R_q}] = R_q$. Then we have

$$\mathrm{var}(\widehat{p}) = \mathrm{var}(\widehat{p_q}) + \underbrace{\mathrm{var}((1-\widehat{p_q})\widehat{R_q})}_{=\blacksquare} + 2\underbrace{\mathrm{cov}(\widehat{p_q}, (1-\widehat{p_q})\widehat{R_q})}_{=\blacktriangle}. \tag{11}$$

We can write the variance ■ and the covariance ▲ as

$$\blacksquare = \mathrm{var}((1-\widehat{p_q})\widehat{R_q}) = (1-p_q)^2\,\mathrm{var}(\widehat{R_q}) + R_q^2\,\mathrm{var}(\widehat{p_q}) + \mathrm{var}(\widehat{p_q})\,\mathrm{var}(\widehat{R_q}),$$
$$\blacktriangle = \mathrm{cov}[\widehat{p_q}, (1-\widehat{p_q})\widehat{R_q}] = -\,\mathrm{var}(\widehat{p_q})R_q,$$

respectively, by exploiting the independence of $\widehat{p_q}$ and $\widehat{R_q}$. By plugging in those expressions in Equation (11) we obtain the result in Equation (3). □

Proof of Proposition 2

Proof.

$$\begin{aligned}
\operatorname{var}(\widetilde{G}) &= \frac{1}{n^2m^2}\operatorname{var}\left(\sum_{i=1}^{n}\sum_{j=1}^{m} g(W_i, Z_{i,j})\right) = \frac{1}{nm^2}\operatorname{var}\left(\sum_{j=1}^{m} g(W_1, Z_{1,j})\right) \\
&= \frac{1}{nm^2}\sum_{j=1}^{m}\sum_{j'=1}^{m}\operatorname{cov}\left(g(W_1, Z_{1,j}), g(W_1, Z_{1,j'})\right) \\
&= \frac{1}{nm^2}\left[m\operatorname{var}(g(W_1, Z_{1,1})) + m(m-1)\operatorname{cov}(g(W_1, Z_{1,1}), g(W_1, Z_{1,2}))\right] \\
&= \frac{1}{nm^2}\left[m\operatorname{var}(g(W_1, Z_{1,1})) + m(m-1)\blacklozenge\right]. \qquad (12)
\end{aligned}$$

where the first equality is a consequence of the independence of $W_1, \ldots, W_n$ and the third equality is a consequence of the independence of $Z_{i,j}$ and $Z_{i,j'}$ conditionally on W_i. Moreover the covariance denoted by $\blacklozenge$ in (12) can be written as follows.

$$\begin{aligned}
\blacklozenge &= \underbrace{\mathbb{E}\big[\operatorname{cov}(g(W_1, Z_{1,1}), g(W_1, Z_{1,2}) \mid W_1)\big]}_{=0 \text{ } Z_{1,1}, Z_{1,2} \text{ independent conditionally on } W_1} + \underbrace{\operatorname{cov}\left(\mathbb{E}[g(W_1, Z_{1,1}) \mid W_1], \mathbb{E}[g(W_1, Z_{1,2}) \mid W_1]\right)}_{=\operatorname{var}(\mathbb{E}[g(W_1, Z_{1,1})|W_1])} \\
&= \operatorname{var}\left(\mathbb{E}[g(W_1, Z_{1,1}) \mid W_1]\right) = \operatorname{var}\left(g(W_1, Z_{1,1})\right) - \mathbb{E}\Big[\operatorname{var}\left(g(W_1, Z_{1,1}) \mid W_1\right)\Big]. \qquad (13)
\end{aligned}$$

Equations (12) and (13) give the result (8).

□

Proof of Corollary 1

Proof. Denote with $e = \beta(A - B),\ f = (\alpha + c)(A - B) + \beta B,\ g = (c + \alpha)B,\ h = C_{\text{tot}} - c_0$, then

$$\operatorname{var}(\widetilde{G})(m) = \frac{em^2 + fm + g}{hm}. \qquad (14)$$

Observe that the first and second derivatives of $\operatorname{var}(\widetilde{G})$ with respect to m are respectively

$$\frac{\partial \operatorname{var}(\widetilde{G})}{\partial m} = \frac{1}{h}\left[e - \frac{g}{m^2}\right], \quad \frac{\partial^2 \operatorname{var}(\widetilde{G})}{\partial m^2} = \frac{2g}{hm^3}.$$

The second derivative is positive for all $m > 0$ then $\operatorname{var}(\widetilde{G})$ is a convex function for $m > 0$ and the point of minimum is equal to the zero of $\partial \operatorname{var}(\widetilde{G})/\partial m$, which is $m = \sqrt{g/e} = \widetilde{m}$.

Since $\mathrm{var}(\widetilde{G})$ is convex in m, the integer that realizes the minimal variance is either $\lfloor \widetilde{m} \rfloor$ or $\lceil \widetilde{m} \rceil$. By plugging in $m = \widetilde{m} - \varepsilon = \sqrt{g/e} - \varepsilon$ and $m = \widetilde{m} - \varepsilon + 1 = \sqrt{g/e} - \varepsilon + 1$ in Equation (14), we obtain the condition in (9). □

Proof of Proposition 3

Proof. First of all notice that the total cost of sampling $\widehat{G}$ is $C_{\text{tot}} = c_0 + n(c + C_{Z|W}) = c_0 n(c + \alpha + \beta)$. By isolating n in the previous equation we obtain $n = \frac{\overline{C}_{\text{tot}}}{c+\alpha+\beta}$, where $\overline{C}_{\text{tot}}$ for the sake of brevity and, by computations similar to those in Proposition 2 we obtain

$$\mathrm{var}(\widehat{G}) = \frac{c + \alpha + \beta}{\overline{C}_{\text{tot}}} \mathrm{var}(g(W_1, Z_{1,1})) = \frac{c + \alpha + \beta}{\overline{C}_{\text{tot}}} A,$$

where $A = \mathrm{var}(g(W_1, Z_{1,1}))$. In the following we will also denote $B = \mathbb{E}\big[\mathrm{var}(g(W_1, Z_{1,1}) \mid W_1)\big]$ as in Corollary 1. Let us now substitute $N_{C_{\text{fix}}}(m^*)$ in equation (8), thus obtaining

$$\begin{aligned}
\mathrm{var}(\widetilde{G}) &= \frac{(c + \alpha + \beta m^*) A m^* - (m^* - 1)(c + \alpha + \beta m^*) B}{\overline{C}_{\text{tot}} m^*} \\
&= \mathrm{var}(\widehat{G}) \frac{(m^*)^2 \beta (A - B) + m^* [(c + \alpha)(A - B) + \beta B] + (c + \alpha) B}{A(c + \alpha + \beta) m^*} \\
&= \mathrm{var}(\widehat{G}) \frac{2(\alpha + c) B + m^* [(c + \alpha)(A - B) + \beta B]}{A(c + \alpha + \beta) m^*}, \qquad (15)
\end{aligned}$$

where in (15) we substituted $(m^*)^2$ from Corollary 1. By rearranging the terms, we obtain

$$\mathrm{var}(\widetilde{G}) = \mathrm{var}(\widehat{G}) \left[1 - \frac{(m^* - 2)(c + \alpha) B + m^* \beta (B - A)}{A(c + \alpha + \beta) m^*} \right] = \mathrm{var}(\widehat{G}) \left[1 - \eta \right].$$

Since $A - B, B, c, \beta, \alpha$ are always positive, then $\eta < 1$ for all $m^* > 0$. Moreover $\eta > 0$ if

$$m^* > \frac{2(\alpha + c) B}{(\alpha + c) B + \beta (A - B)}.$$

□

References

Abrahamson, I. (1964). Orthant probabilities for the quadrivariate normal distribution. *The Annals of Mathematical Statistics*, 35(4):1685–1703.

Bayarri, M. J., Berger, J. O., Calder, E. S., Dalbey, K., Lunagomez, S., Patra, A. K., Pitman, E. B., Spiller, E. T., and Wolpert, R. L. (2009). Using statistical and computer models to quantify volcanic hazards. *Technometrics*, 51(4):402–413.

Bect, J., Ginsbourger, D., Li, L., Picheny, V., and Vazquez, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22 (3):773–793.

Bolin, D. and Lindgren, F. (2015). Excursion and contour uncertainty regions for latent Gaussian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(1):85–106.

Botev, Z. I. (2017). The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79:1–24.

Bratley, P. and Fox, B. L. (1988). Algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Trans. Math. Software*, 14(1):88–100.

Chevalier, C. (2013). *Fast uncertainty reduction strategies relying on Gaussian process models*. PhD thesis, University of Bern.

Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., and Richet, Y. (2014a). Fast kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465.

Chevalier, C., Ginsbourger, D., Bect, J., and Molchanov, I. (2013). Estimating and quantifying uncertainties on level sets using the Vorob'ev expectation and deviation with Gaussian process models. In Uciński, D., Atkinson, A., and Patan, C., editors, *mODa 10 Advances in Model-Oriented Design and Analysis*. Physica-Verlag HD.

Chevalier, C., Picheny, V., and Ginsbourger, D. (2014b). The KrigInv package: An efficient and user-friendly R implementation of kriging-based inversion algorithms. *Computational Statistics and Data Analysis*, 71:1021–1034.

Cox, D. R. and Wermuth, N. (1991). A simple approximation for bivariate and trivariate normal integrals. *International Statistical Review*, 59(2):263–269.

Dickmann, F. and Schweizer, N. (2016). Faster comparison of stopping times by nested conditional Monte Carlo. *Journal of Computational Finance*, pages 101–123.

French, J. P. and Sain, S. R. (2013). Spatio-temporal exceedance locations and confidence regions. *Annals of Applied Statistics*, 7(3):1421–1449.

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(2):141–149.

Genz, A. and Bretz, F. (2002). Comparison of methods for the computation of multivariate t probabilities. *Journal of Computational and Graphical Statistics*, 11(4):950–971.

Genz, A. and Bretz, F. (2009). *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics 195. Springer-Verlag.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., Bornkamp, B., and Hothorn, T. (2012). *mvtnorm: Multivariate Normal and t Distributions*. R package version 0.9-9992.

Geweke, J. (1991). Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities. In *Computing science and statistics: Proceedings of the 23rd symposium on the interface*, pages 571–578.

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(1):85–134.

Hajivassiliou, V. A. and McFadden, D. L. (1998). The method of simulated scores for the estimation of LDV models. *Econometrica*, 66(4):863–896.

Hammersley, J. (1956). Conditional Monte Carlo. *Journal of the ACM (JACM)*, 3(2):73–76.

Hammersley, J. and Morton, K. (1956). A new monte carlo technique: antithetic variates. In *Mathematical proceedings of the Cambridge philosophical society*, volume 52, pages 449–475. Cambridge Univ Press.

Horrace, W. C. (2005). Some results on the multivariate truncated normal distribution. *Journal of Multivariate Analysis*, 94(1):209–221.

Kahn, H. (1950). Random sampling (Monte Carlo) techniques in neutron attenuation problems–I. *Nucleonics*, 6(5):27–33.

Kahn, H. and Marshall, A. W. (1953). Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278.

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106.

Lemieux, C. (2009). *Monte Carlo and quasi-Monte Carlo sampling*. Springer.

Miwa, T., Hayter, A., and Kuriki, S. (2003). The evaluation of general non-centred orthant probabilities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):223–234.

Molchanov, I. (2005). *Theory of Random Sets*. Springer, London.

Moran, P. (1984). The monte carlo evaluation of orthant probabilities for multivariate normal distributions. *Australian Journal of Statistics*, 26(1):39–44.

Owen, D. B. (1956). Tables for computing bivariate normal probabilities. *The Annals of Mathematical Statistics*, 27(4):1075–1090.

Picheny, V., Ginsbourger, D., Richet, Y., and Caplin, G. (2013). Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55(1):2–13.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*. MIT Press.

Ridgway, J. (2016). Computation of gaussian orthant probabilities in high dimension. *Statistics and Computing*, 26(4):899–916.

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer.

Robert, C. P. (1995). Simulation of truncated normal variables. *Statistics and Computing*, 5(2):121–125.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435.

Schervish, M. J. (1984). Algorithm AS 195: Multivariate normal probabilities with error bound. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 33(1):81–94.

Tong, Y. L. (2012). *The multivariate normal distribution*. Springer.