JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2022, VOL. 31, NO. 3, 709–720
https://doi.org/10.1080/10618600.2022.2036614

Taylor & Francis
Taylor & Francis Group

Scalable Computation of Predictive Probabilities in Probit Models with Gaussian Process Priors

Jian Cao [a], Daniele Durante[b], and Marc G. Genton[a]

[a]Statistics Program, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia; [b]Department of Decision Sciences and Bocconi Institute for Data Science and Analytics, Bocconi University, Milano, Italy

ABSTRACT

Predictive models for binary data are fundamental in various fields, and the growing complexity of modern applications has motivated several flexible specifications for modeling the relationship between the observed predictors and the binary responses. A widely-implemented solution is to express the probability parameter via a probit mapping of a Gaussian process indexed by predictors. However, unlike for continuous settings, there is a lack of closed-form results for predictive distributions in binary models with Gaussian process priors. Markov chain Monte Carlo methods and approximation strategies provide common solutions to this problem, but state-of-the-art algorithms are either computationally intractable or inaccurate in moderate-to-high dimensions. In this article, we aim to cover this gap by deriving closed-form expressions for the predictive probabilities in probit Gaussian processes that rely either on cumulative distribution functions of multivariate Gaussians or on functionals of multivariate truncated normals. To evaluate these quantities we develop novel scalable solutions based on tile-low-rank Monte Carlo methods for computing multivariate Gaussian probabilities, and on mean-field variational approximations of multivariate truncated normals. Closed-form expressions for the marginal likelihood and for the posterior distribution of the Gaussian process are also discussed. As shown in simulated and real-world empirical studies, the proposed methods scale to dimensions where state-of-the-art solutions are impractical.

ARTICLE HISTORY
Received September 2020
Revised November 2021

KEYWORDS
Binary data; Gaussian process; Multivariate truncated normal; Probit model; Unified skew-normal; Variational Bayes

1. Introduction

There is an increasing demand in various fields of application for flexible models that can accurately characterize complex relations among a vector of binary response data $\mathbf{y} = (y_1, \ldots, y_n)^\intercal$ and a set of predictors $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal$, where $y_i \in \{0; 1\}$, whereas $\mathbf{x}_i = (x_{i1}, \ldots, x_{iq})^\intercal \in \mathbb{R}^q$, for every unit $i = 1, \ldots, n$. Common solutions address this goal by replacing the linear predictor $\mathbf{X}\boldsymbol{\beta} = (\mathbf{x}_1^\intercal\boldsymbol{\beta}, \ldots, \mathbf{x}_n^\intercal\boldsymbol{\beta})^\intercal \in \mathbb{R}^n$ within the generalized linear model for $\mathbf{y}$ (Nelder and Wedderburn 1972) with a more flexible vector

$$\mathbf{f}(\mathbf{X}) = \big[f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\big]^\intercal \in \mathbb{R}^n,$$

which accounts for complex nonlinear relationships between the response and the predictors, thus, enhancing predictive power. Notable examples of this approach within the Bayesian setting define $\mathbf{f}(\mathbf{X})$ via additive trees (Chipman, George, and McCulloch 2010), Bayesian P–splines (Brezger and Lang 2006) and Gaussian processes (GP) (Rasmussen and Williams 2006), among others.

Motivated by the success of GP for classification (Neal 1999; Opper and Winther 2000; De Oliveira 2005; Chu and Ghahramani 2005; Kuss and Rasmussen 2005; Girolami and Rogers 2006; Rasmussen and Williams 2006; Choudhuri, Ghosal, and Roy 2007; Riihimäki, Jylänki, and Vehtari 2013), we aim at deriving improved methods to evaluate predictive probabilities within this class of models under the probit link. Following the standard practice, we assume that y_i, for $i = 1, \ldots, n$, are conditionally independent realizations from Bernoulli variables with probabilities $\Phi(f(\mathbf{x}_i)) = \text{pr}(y_i = 1 \mid f(\mathbf{x}_i))$, $i = 1, \ldots, n$, where $\Phi(f(\mathbf{x}))$ is the cumulative distribution function of a standard Gaussian evaluated at $f(\mathbf{x})$, whereas $f(\mathbf{x})$ is assigned a GP prior with mean function $m(\mathbf{x}) = \mathbb{E}(f(\mathbf{x}))$ and covariance kernel $K(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$. In routine implementations (e.g., Kuss and Rasmussen 2005; Rasmussen and Williams 2006), $K(\mathbf{x}, \mathbf{x}')$ denotes a prespecified function indexed by a low-dimensional vector of hyperparameters $\boldsymbol{\alpha} \in \mathbb{R}^d$, where $d \in \{1; 2; 3\}$ in commonly implemented covariance functions (Rasmussen and Williams 2006, chap. 4.2). These quantities can be either fixed to default values by inheriting guidelines from Bayesian regression for binary data (Gelman 2008; Chopin and Ridgway 2017), or can be estimated leveraging information from observed data via direct maximization of the marginal likelihood (e.g., Kuss and Rasmussen 2005; Rasmussen and Williams 2006); see Section 5 for a discussion on estimation of $\boldsymbol{\alpha}$ in large d settings. The mean function $m(\mathbf{x})$ is, instead, commonly set equal to 0, or is assigned a further layer of hierarchy which typically specifies $m(\mathbf{x})$ via a linear combination $\mathbf{x}^\intercal\boldsymbol{\beta}$ of the predictors $\mathbf{x}$, where $\boldsymbol{\beta}$ denotes a q-dimensional vector of coefficients generally assumed to have independent

CONTACT Jian Cao jian.cao@tamu.edu Statistics Program, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia.

Gaussian priors $N(0, \delta^2)$ (e.g., Rasmussen and Williams 2006, chap. 2.7). Although estimation and uncertainty quantification for $\boldsymbol{\beta}$ can be of interest, the key aim of this article is to improve predictive inference in probit GPs. Such a goal is in line with the general focus of GP literature that often employs Gaussian process representations to improve predictive performance relative to classical linear regression models (e.g., Kuss and Rasmussen 2005; Rasmussen and Williams 2006; Girolami and Rogers 2006; Nickisch and Rasmussen 2008; Riihimäki, Jylänki, and Vehtari 2013). Consistent with this goal, when $\mathbf{x}^\intercal\boldsymbol{\beta}$ enters the GP mean function, we follow Rasmussen and Williams (2006, chap. 2.7) by marginalizing out $\boldsymbol{\beta}$ and evaluating predictive probabilities under the induced GP prior for $f(\mathbf{x})$, with mean function equal to 0 and covariance kernel given by $K(\mathbf{x}, \mathbf{x}') + \delta^2\mathbf{x}^\intercal\mathbf{x}'$. As discussed in Rasmussen and Williams (2006, chap. 2.7), this updated kernel formally allows to fully exploit possible linear relationships among the response and covariates in predictive inference.

Leveraging basic GP properties (Rasmussen and Williams 2006) and assuming, without any loss of generality, no overlap in $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the aforementioned probit Gaussian process models can be generally expressed as

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{f}(\mathbf{X})) &= \prod_{i=1}^{n} \Phi(f(\mathbf{x}_i))^{y_i}[1 - \Phi(f(\mathbf{x}_i))]^{1-y_i}, \\ p(\mathbf{f}(\mathbf{X})) &= \phi_n(\mathbf{f}(\mathbf{X}) - \boldsymbol{\xi}; \boldsymbol{\Omega}), \end{aligned} \tag{1}$$

where $\phi_n(\mathbf{f}(\mathbf{X}) - \boldsymbol{\xi}; \boldsymbol{\Omega})$ denotes the density function of a multivariate Gaussian distribution $N_n(\boldsymbol{\xi}, \boldsymbol{\Omega})$ for $\mathbf{f}(\mathbf{X})$, with mean vector $\boldsymbol{\xi} = [m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n)]^\intercal$, and $n \times n$ covariance matrix $\boldsymbol{\Omega}$ having entries $\boldsymbol{\Omega}_{i,i'} = K(\mathbf{x}_i, \mathbf{x}_{i'})$, for every $i = 1, \ldots, n$ and $i' = 1, \ldots, n$. Model (1) has attracted a considerable interest due to its flexibility and its direct connection with binary discrete choice models based on Gaussian latent utilities $z_i = f(\mathbf{x}_i) + \varepsilon_i$, with $\varepsilon_i \sim N(0, 1)$, independently for $i = 1, \ldots, n$ (Albert and Chib 1993). In fact, $\text{pr}(y_i = 1 \mid f(\mathbf{x}_i)) = \Phi(f(\mathbf{x}_i)) = \text{pr}(z_i > 0 \mid f(\mathbf{x}_i))$. In such settings, a main goal of inference is to evaluate the predictive probabilities of new responses y_{n+1} at a given point $\mathbf{x}_{n+1}$. Recalling Rasmussen and Williams (2006, chap. 3.3), such quantities can be defined as

$$\begin{aligned} &\text{pr}(y_{n+1} = 1 \mid \mathbf{y}) = 1 - \text{pr}(y_{n+1} = 0 \mid \mathbf{y}) \\ &= \int \Phi(f(\mathbf{x}_{n+1})) \left[\int p(f(\mathbf{x}_{n+1}), \mathbf{f}(\mathbf{X}) \mid \mathbf{y}) \text{d}\mathbf{f}(\mathbf{X}) \right] \text{d}f(\mathbf{x}_{n+1}), \end{aligned} \tag{2}$$

where $p(f(\mathbf{x}_{n+1}), \mathbf{f}(\mathbf{X}) \mid \mathbf{y})$ is the joint posterior density of $(f(\mathbf{x}_{n+1}), \mathbf{f}(\mathbf{X}))$ induced by model (1), which does not seem to have an obvious closed form due to the apparent absence of conjugacy between the probit likelihood and the multivariate Gaussian prior for $(f(\mathbf{x}_{n+1}), \mathbf{f}(\mathbf{X}))$ under (1). This has motivated extensive research to compute the predictive probabilities in probit models with multivariate Gaussian priors either via Monte Carlo methods relying on samples from $p(f(\mathbf{x}_{n+1}), \mathbf{f}(\mathbf{X}) \mid \mathbf{y})$ (Albert and Chib 1993; Neal 1999; De Oliveira 2005; Holmes and Held 2006; Choudhuri, Ghosal, and Roy 2007; Pakman and Paninski 2014; Durante 2019) or by deriving tractable approximations of $p(f(\mathbf{x}_{n+1}), \mathbf{f}(\mathbf{X}) \mid \mathbf{y})$ (Kuss and Rasmussen 2005; Chu and Ghahramani 2005; Girolami and Rogers 2006; Rasmussen and Williams 2006; Consonni and Marin 2007; Nickisch and Rasmussen 2008; Riihimäki, Jylänki, and Vehtari 2013) that allow simple evaluation of (2). Such methods provide state-of-the-art solutions in small-to-moderate dimensional settings, but tend to become inaccurate or computationally impractical in higher dimensions (Chopin and Ridgway 2017; Johndrow et al. 2019; Durante 2019; Fasano, Durante, and Zanella in press). This issue is inherent to probit GPs where, by definition, the dimension of $\mathbf{f}(\mathbf{X})$ is n, or slightly lower when there is overlap in locations, with n being relatively large in most studies.

In this article we aim to cover the above gap by providing novel closed-form expressions for the predictive probabilities in probit GPs along with improved methods to evaluate the involved quantities in high dimensions. More specifically, in Section 2.1 we first derive a closed-form expression for the marginal likelihood $p(\mathbf{y})$ under model (1), and then exploit this result to show that $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$ can be expressed as the ratio between cumulative distribution functions of multivariate Gaussians with dimensions $n + 1$ and n, respectively. To overcome the known issues associated with the evaluation of these two quantities in high dimensions (Chopin 2011; Botev 2017; Cao et al. 2019, 2021) we introduce an error-reduction technique for computing ratios of Gaussian cumulative distribution functions that builds on the tile-low-rank method in Cao et al. (2021), and substantially reduces the computational time of state-of-the-art strategies such as minimax tilting methods (Botev 2017) and Hamiltonian Monte Carlo samplers (STAN) (Hoffman and Gelman 2014), without affecting accuracy. In Section 2.2, we further derive an alternative representation of $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$, which relies on functionals of multivariate truncated normals, and we address the intractability of such variables in high dimensions by proposing a variational approximation based on univariate truncated normals which allows accurate and computationally tractable evaluation of predictive probabilities in high-dimensional contexts. As clarified in Section 2.2, this solution is computationally more scalable than currently implemented expectation-propagation approximations (e.g., Kuss and Rasmussen 2005; Chu and Ghahramani 2005; Riihimäki, Jylänki, and Vehtari 2013), and improves the accuracy of routinely used variational solutions (e.g., Girolami and Rogers 2006), that commonly rely on more restrictive mean-field assumptions, than those required under the proposed approximation. These results are also related to the conditional distribution of the GP given the binary responses, which we show to coincide with a unified skew-normal (SUN) (Arellano-Valle and Azzalini 2006) by adapting recent results in Durante (2019) on Bayesian probit regression. The magnitude of the improvements provided by the new methods presented in Sections 2.1–2.2 relative to state-of-the-art competitors is illustrated in simulations in Section 3, and in an environmental application to Saudi Arabia windspeed in Section 4. Section 5 contains concluding remarks, whereas all the proofs can be found in the Appendix. Complete R code to implement the proposed methods and quantify the improvements relative to state-of-the-art competitors in simulation studies is available at *https://github.com/danieledurante/PredProbitGP*.

2. Improved Evaluation of Predictive Probabilities in Probit Gaussian Processes

Sections 2.1 and 2.2 present novel expressions for the predictive probabilities in probit GPs along with improved methods to evaluate the involved quantities efficiently in high dimensions. Feasible grid strategies to estimate the GP hyperparameters $\boldsymbol{\alpha}$ are also proposed; see Section 5 for a discussion on the computational tractability of these routines in relation to the dimension of $\boldsymbol{\alpha}$.

2.1. Evaluation via Gaussian Probability Ratios

To introduce the closed-form expression for $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$ based on ratios of multivariate Gaussian cumulative distribution functions, first note that by leveraging known properties of Gaussian variables, the probit likelihood in (1) can be written as

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{f}(\mathbf{X})) &= \prod_{i=1}^{n} \Phi(f(\mathbf{x}_i))^{y_i}[1 - \Phi(f(\mathbf{x}_i))]^{1-y_i} \\ &= \prod_{i=1}^{n} \Phi[(2y_i - 1)f(\mathbf{x}_i)] = \Phi_n(\mathbf{D}\mathbf{f}(\mathbf{X}); \mathbf{I}_n), \end{aligned}$$

where $\Phi_n(\mathbf{D}\mathbf{f}(\mathbf{X}); \mathbf{I}_n)$ is the cumulative distribution function of a zero-mean n-variate Gaussian with identity covariance matrix $\mathbf{I}_n$, evaluated at $\mathbf{D}\mathbf{f}(\mathbf{X})$, with $\mathbf{D} = \text{diag}[(2y_1 - 1), \ldots, (2y_n - 1)]$. Leveraging this form and adapting results in Lemma 7.1 of Azzalini and Capitanio (2014) to our setting, we can easily express the marginal likelihood under model (1) as

$$\begin{aligned} p(\mathbf{y}) &= \int \Phi_n(\mathbf{D}\mathbf{f}(\mathbf{X}); \mathbf{I}_n)\phi_n(\mathbf{f}(\mathbf{X}) - \boldsymbol{\xi}; \boldsymbol{\Omega})\text{d}\mathbf{f}(\mathbf{X}) \\ &= \Phi_n(\mathbf{D}\boldsymbol{\xi}; \mathbf{I}_n + \mathbf{D}\boldsymbol{\Omega}\mathbf{D}^\intercal). \end{aligned} \tag{3}$$

As it will be discussed later on in this article, Equation (3) provides a closed-form expression that can be useful to estimate the GP hyperparameters $\boldsymbol{\alpha}$ via direct maximization of $p(\mathbf{y})$. In addition, as shown in Proposition 1, Equation (3) also allows to derive closed-form expressions for $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$.

Proposition 1. Under model (1), the predictive probability for a new binary response $y_{n+1} \in \{0; 1\}$ with predictor $\mathbf{x}_{n+1} \in \mathbb{R}^q$ is

$$\begin{aligned} \text{pr}(y_{n+1} = 1 \mid \mathbf{y}) &= 1 - \text{pr}(y_{n+1} = 0 \mid \mathbf{y}) \\ &= \frac{\Phi_{n+1}(\mathbf{D}^*\boldsymbol{\xi}^*; \mathbf{I}_{n+1} + \mathbf{D}^*\boldsymbol{\Omega}^*\mathbf{D}^{*\intercal})}{\Phi_n(\mathbf{D}\boldsymbol{\xi}; \mathbf{I}_n + \mathbf{D}\boldsymbol{\Omega}\mathbf{D}^\intercal)}, \end{aligned} \tag{4}$$

with $\boldsymbol{\xi}^* = [\boldsymbol{\xi}^\intercal, m(\mathbf{x}_{n+1})]^\intercal$, $\mathbf{D}^* = \text{diag}[(2y_1 - 1), \ldots, (2y_n - 1), 1]$, whereas $\boldsymbol{\Omega}^*$ is obtained by including one additional row and column to $\boldsymbol{\Omega}$, which are defined as $\boldsymbol{\Omega}^{*\intercal}_{[n+1,\cdot]} = \boldsymbol{\Omega}^*_{[\cdot,n+1]} = [K(\mathbf{x}_{n+1}, \mathbf{x}_1), \ldots, K(\mathbf{x}_{n+1}, \mathbf{x}_n), K(\mathbf{x}_{n+1}, \mathbf{x}_{n+1})]^\intercal$.

In order to prove Proposition 1, it is sufficient to notice that, by the Bayes' rule, $\text{pr}(y_{n+1} = 1 \mid \mathbf{y}) = \text{p}(y_{n+1} = 1, \mathbf{y})/p(\mathbf{y})$ where $\text{p}(y_{n+1} = 1, \mathbf{y})$ and $p(\mathbf{y})$ are the marginal likelihoods of $(y_{n+1} = 1, \mathbf{y})$ and $\mathbf{y}$, respectively, under model (1). Replacing such quantities with their closed-form expression as in (3), leads to (4). See Appendix for a more detailed proof which also includes additional clarifications on Equation (3).

The evaluation of (4) requires the calculation of cumulative distribution functions of multivariate Gaussians, which is known to be a challenging task in high dimensions (Genz 1992; Chopin 2011; Botev 2017; Genton, Keyes, and Turkiyyah 2018; Cao et al. 2019, 2021). Recent advances via minimax tilting (Botev 2017) allow accurate evaluation of such quantities, but face an increased computational cost which makes these strategies rapidly impractical as n grows. A possible solution to such an issue can be found in the separation-of-variable (SOV) algorithm originally introduced by Genz (1992), and subsequently improved in terms of scalability by Cao et al. (2021). Such a routine decomposes the generic multivariate Gaussian probability $\Phi_n(\mathbf{a}, \mathbf{b}; \boldsymbol{\Sigma}) = \int_{\mathbf{a}}^{\mathbf{b}} \phi_n(\mathbf{u}; \boldsymbol{\Sigma})\text{d}\mathbf{u}$ as

$$\begin{aligned} \Phi_n(\mathbf{a}, \mathbf{b}; \boldsymbol{\Sigma}) &= (e_1 - d_1)\int_0^1 (e_2 - d_2)\cdots\int_0^1 (e_n - d_n)\int_0^1 \text{d}\mathbf{w}_{-n} \\ &= \mathbb{E}_{\mathbf{w}_{-n}}[(e_1 - d_1)\cdots(e_n - d_n)] \\ &= \mathbb{E}_{\mathbf{w}_{-n}}\left[\prod_{i=1}^{n}(e_i - d_i)\right], \end{aligned} \tag{5}$$

with $\mathbf{w}_{-n} = (w_1, \ldots, w_{n-1})^\intercal$ denoting a vector of uniform entries $w_j \sim \text{U}(0, 1)$, for $j = 1, \ldots, n-1$, whereas

$$\begin{aligned} d_i &= \Phi\left(\left[a_i - \sum_{j=1}^{i-1} l_{ij}\Phi^{-1}[d_j + w_j(e_j - d_j)]\right] l_{ii}^{-1}\right), \\ e_i &= \Phi\left(\left[b_i - \sum_{j=1}^{i-1} l_{ij}\Phi^{-1}[d_j + w_j(e_j - d_j)]\right] l_{ii}^{-1}\right), \end{aligned}$$

for $i = 1, \ldots, n$, where l_{ij} is the (ij)-th coefficient in the lower Cholesky factor of $\boldsymbol{\Sigma}$. This decomposition transforms the integration region into the unit hypercube, thus, allowing the evaluation of $\Phi_n(\mathbf{a}, \mathbf{b}; \boldsymbol{\Sigma})$ via functionals of uniform densities. To further improve the quality of the above estimator, more recent implementations (Trinh and Genz 2015) combine (5) with a univariate reordering preconditioner that rearranges the integration variables and produces the corresponding Cholesky factor simultaneously at the same $\mathcal{O}(n^3)$ cost of the Cholesky factorization. This prioritization strategy accounts for the width of the integration limits by reordering the variables to ensure that those having smallest expected values appear as outermost integration variables. Such approach is shown in Trinh and Genz (2015) and Cao et al. (2021) to improve the Monte Carlo convergence rate of (5), whose integrand is evaluated R times—corresponding to the Monte Carlo sample size—each of which has a cost of $\mathcal{O}(n^2)$. Such costs allow the implementation of this strategy in settings with $n \leq 1000$, thus, motivating more scalable options in high dimensions. Cao et al. (2021) address this issue via a tile-low-rank representation for $\boldsymbol{\Sigma}$ that reduces the cost of the SOV algorithm by substituting the dense matrix-vector multiplication with the low-rank matrix-vector multiplication. A compatible block-reordering is also introduced in place of the univariate reordering to improve the convergence rate at the same cost of the low-rank Cholesky factorization. Specifically, the block-reordering orders the integration variables on the block level based on crude estimates of the block-wise marginal probabilities as shown in Figure 1. Both the block-reordering and the tile-low-rank version of the SOV algorithm reach their optimal complexities of $\mathcal{O}(n^{5/2})$ and $\mathcal{O}(n^{3/2})$, respectively, when the block size in the tile-low-rank representation is $n^{1/2}$, thus, reducing the computational complexity of the classical SOV algorithm by $n^{1/2}$, and allowing implementation in tens of thousands of dimensions.

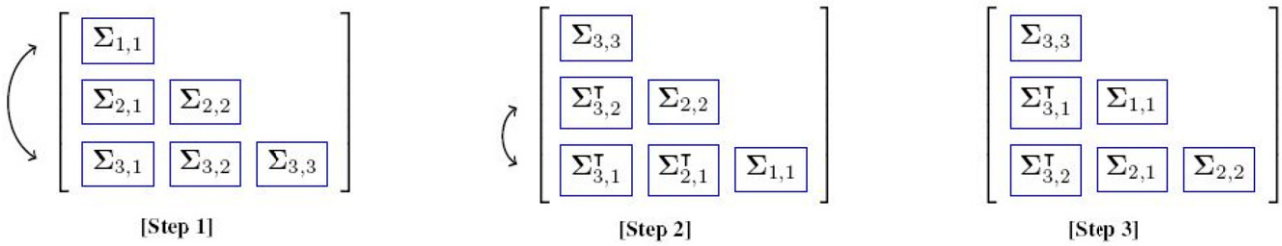

Figure 1. Illustration of the block-reordering strategy (Cao et al. 2021). [Step 1]: Compute $\min\{\Phi_{n_1}(\mathbf{a}_1, \mathbf{b}_1; \boldsymbol{\Sigma}_{1,1}); \Phi_{n_2}(\mathbf{a}_2, \mathbf{b}_2; \boldsymbol{\Sigma}_{2,2}); \Phi_{n_3}(\mathbf{a}_3, \mathbf{b}_3; \boldsymbol{\Sigma}_{3,3})\}$, and suppose that the solution is $\Phi_{n_3}(\mathbf{a}_3, \mathbf{b}_3; \boldsymbol{\Sigma}_{3,3})$. Then, switch first and third block rows and columns, and perform univariate reordering for $\boldsymbol{\Sigma}_{3,1}$, $\boldsymbol{\Sigma}_{3,2}$ and $\boldsymbol{\Sigma}_{3,3}$. [Step 2]: Compute $\min\{\Phi_{n_1}(\mathbf{a}_1, \mathbf{b}_1; \boldsymbol{\Sigma}_{1,1}); \Phi_{n_2}(\mathbf{a}_2, \mathbf{b}_2; \boldsymbol{\Sigma}_{2,2})\}$, and suppose that the solution is $\Phi_{n_1}(\mathbf{a}_1, \mathbf{b}_1; \boldsymbol{\Sigma}_{1,1})$. Then, switch second and third block rows and columns, and perform univariate reordering for $\boldsymbol{\Sigma}_{3,1}$, $\boldsymbol{\Sigma}_{2,1}$ and $\boldsymbol{\Sigma}_{1,1}$. [Step 3]: Perform univariate reordering for $\boldsymbol{\Sigma}_{2,1}$, $\boldsymbol{\Sigma}_{3,2}$ and $\boldsymbol{\Sigma}_{2,2}$.

Algorithm 1: Compute (4) via the estimator (7)

[a] Set $\mathbf{a} = -\infty, \mathbf{b} = \mathbf{D}^*\boldsymbol{\xi}^*, \boldsymbol{\Sigma} = \mathbf{I}_{n+1} + \mathbf{D}^*\boldsymbol{\Omega}^*\mathbf{D}^{*\intercal}$, and draw $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(R)}$ uniform samples from the unit hypercurbe in $(0,1)^n$.
[b] Apply block-reordering (Cao et al. 2021) to $(\mathbf{a}_{-(n+1)}, \mathbf{b}_{-(n+1)}, \boldsymbol{\Sigma}_{-(n+1)})$, which produces the tile-low-rank Cholesky factor $\mathbf{L}_{-(n+1)}$, and the reordered $\mathbf{a}_{-(n+1)}$ and $\mathbf{b}_{-(n+1)}$.
[c] Compute $\mathbf{L}_{[n+1,1:n+1]}$ using $\boldsymbol{\Sigma}$ and $\mathbf{L}_{-(n+1)}$.
[d] Obtain the quantities required to evaluate Equation (7).
for $r = 1, \ldots, R$ **do**
  [d.1] Compute $e_i(\mathbf{w}_{-n}^{(r)}) - d_i(\mathbf{w}_{-n}^{(r)})$, for every statistical unit $i = 1, \ldots, n$, by applying the tile-low-rank variant of (5) (Cao et al. 2021) to $(\mathbf{a}_{-(n+1)}, \mathbf{b}_{-(n+1)}, \boldsymbol{\Sigma}_{-(n+1)})$.
  Store also the vector $\mathbf{v}^{(r)} = [\Phi^{-1}(d_1(\mathbf{w}_{-n}^{(r)}) + w_1^{(r)}[e_1(\mathbf{w}_{-n}^{(r)}) - d_1(\mathbf{w}_{-n}^{(r)})]), \ldots, \Phi^{-1}(d_n(\mathbf{w}_{-n}^{(r)}) + w_n^{(r)}[e_n(\mathbf{w}_{-n}^{(r)}) - d_n(\mathbf{w}_{-n}^{(r)})])]^{\intercal}$.
  [d.2] Set $e_{n+1}(\mathbf{w}^{(r)}) - d_{n+1}(\mathbf{w}^{(r)}) = \Phi(\frac{b_{n+1} - \mathbf{L}_{[n+1,1:n]}\mathbf{v}^{(r)}}{l_{n+1,n+1}}) - \Phi(\frac{a_{n+1} - \mathbf{L}_{[n+1,1:n]}\mathbf{v}^{(r)}}{l_{n+1,n+1}})$.
[e] Estimate (4) via Monte Carlo as in equation (7) using the quantities computed in step **[d]**.

Although these techniques can be effectively implemented to evaluate multivariate Gaussian probabilities as in (3), the calculation of ratios among such quantities as in (4) typically requires higher accuracy. Unfortunately, as discussed in Botev (2017) and Cao et al. (2021), the estimation errors of tail multivariate Gaussian probabilities, that also include the cumulative distribution function, can be as large as the probability estimates themselves when n is in hundreds to thousands of dimensions, thus, producing unreliable ratio estimates. To address this issue, we propose an error-reduction technique that avoids computing the numerator and the denominator in (4) separately, but combines their evaluation under the tile-low-rank representation. Indeed, as is clear from Proposition 1, the denominator in (4) coincides with the numerator without the last integration variable. Hence, keeping the general notation of the SOV algorithm and leveraging (5), expression (4) can be rewritten in the generic form

$$\frac{\Phi_{n+1}(\mathbf{a}, \mathbf{b}; \boldsymbol{\Sigma})}{\Phi_n(\mathbf{a}_{-(n+1)}, \mathbf{b}_{-(n+1)}; \boldsymbol{\Sigma}_{-(n+1)})} = \frac{\mathbb{E}_{\mathbf{w}}([\prod_{i=1}^n (e_i - d_i)] \cdot (e_{n+1} - d_{n+1}))}{\mathbb{E}_{\mathbf{w}_{-n}}[\prod_{i=1}^n (e_i - d_i)]}, \tag{6}$$

where e_i and d_i are defined as in Equation (5) for $i = 1, \ldots, n+1$, whereas $\mathbf{a}_{-(n+1)}$, $\mathbf{b}_{-(n+1)}$ and $\mathbf{w}_{-n}$ are obtained by removing the $(n+1)$-th element in both $\mathbf{a}$ and $\mathbf{b}$, and the n-th entry in $\mathbf{w}$, respectively. Similarly, $\boldsymbol{\Sigma}_{-(n+1)}$ coincides with $\boldsymbol{\Sigma}$ without the $(n+1)$-th row and column. As is clear from (6), the quantities $(e_1 - d_1), \ldots, (e_n - d_n)$ are the same deterministic functions of $\mathbf{w}$ both in the numerator and in the denominator, and hence, using the same set of Monte Carlo samples $\mathbf{w}$ in the n-dimensional hypercube for estimating the two expectations could significantly reduce the estimation error of their ratio. In particular, our proposed ratio estimator is

$$\hat{\text{pr}}(y_{n+1} = 1 \mid \mathbf{y}) = \frac{\frac{1}{R}\sum_{r=1}^R ([\prod_{i=1}^n (e_i^{(r)} - d_i^{(r)})] \cdot (e_{n+1}^{(r)} - d_{n+1}^{(r)}))}{\frac{1}{R}\sum_{r=1}^R [\prod_{i=1}^n (e_i^{(r)} - d_i^{(r)})]}, \tag{7}$$

where the generic quantities $e_i^{(r)} = e_i(\mathbf{w}^{(r)})$ and $d_i^{(r)} = d_i(\mathbf{w}^{(r)})$ denote the values of e_i and d_i in (5) evaluated at the Monte Carlo sample $\mathbf{w}^{(r)}$ of $\mathbf{w}$. Hence, $e_i^{(r)} = e_i(\mathbf{w}_{-n}^{(r)})$ and $d_i^{(r)} = d_i(\mathbf{w}_{-n}^{(r)})$ for every $i = 1, \ldots, n$, whereas for unit $n+1$ these quantities are defined as $e_{n+1}^{(r)} = e_{n+1}(\mathbf{w}^{(r)})$ and $d_{n+1}^{(r)} = d_{n+1}(\mathbf{w}^{(r)})$. Estimator (7) is asymptotically unbiased because the numerator and the denominator converge to $\mathbb{E}_{\mathbf{w}}[(e_1 - d_1) \ldots (e_n - d_n) \cdot (e_{n+1} - d_{n+1})]$ and $\mathbb{E}_{\mathbf{w}_{-n}}[(e_1 - d_1) \ldots (e_n - d_n)]$, respectively, and hence, Equation (7) converges to (6) in probability. Moreover, Equation (7) is guaranteed to be in $(0,1)$, thus, producing an estimator whose variance is always smaller than 0.25. This is not the case when the numerator and the denominator in (4) are estimated separately. Indeed, as discussed in Botev (2017) and Cao et al. (2021), when n is high the estimation errors of the two cumulative distribution functions in (4) are often as large as the estimates themselves, thus, producing estimated ratios possibly outside of the range $(0,1)$, and with high variance.

The pseudo-code to evaluate (4) via the estimator presented in (7) is provided in Algorithm 1. In step **[b]** of Algorithm 1, the block-reordering produces a new variable order which is used to reorder the integration limits, whereas in step **[c]** the inverse matrices of the diagonal blocks of the tile-low-rank Cholesky factor computed in step **[b]** are recycled to maximize efficiency. Also the quantities in **[d.1]** do not need to be re-evaluated every time a new prediction is required since

they only depend on the observed training data, and hence, such quantities can be precomputed and stored separately. This yields an overall computational complexity of $\mathcal{O}(n^{5/2} + Rn^{3/2})$ for Algorithm 1, which comprises the $\mathcal{O}(n^{5/2})$ precomputation cost of the block-reordering strategy to produce the tile-low-rank Cholesky factor, and the $\mathcal{O}(n^{3/2})$ operations per sample to compute the quantities in step **[d]**. This allows to reduce the overall complexity of other state-of-the-art accurate alternatives for evaluating (4), such as the strategy proposed by Botev (2017), that has an $\mathcal{O}(n^3)$ precomputation cost for obtaining the minimax exponentially-tilted estimate, and then requires $\mathcal{O}(n^2)$ matrix-vector multiplication operations per sample, for a total of $\mathcal{O}(n^3 + Rn^2)$.

The computational gains achieved under Algorithm 1 are also inherited when adapting the method in Cao et al. (2021) to evaluate the marginal likelihood in Equation (3), thereby facilitating the development of feasible estimation strategies for the GP hyperparameters $\boldsymbol{\alpha}$ via the maximization of $p(\mathbf{y})$. Although this task is amenable to a variety of gradient-based optimization algorithms, in practice, the implementation of these routines, might be subject to computational bottlenecks and tedious calculations which involve derivatives of multivariate Gaussian cumulative distribution functions. To circumvent these issues, we propose to rely on a heuristic grid search strategy which evaluates $p(\mathbf{y})$ at several reasonable combinations of $\boldsymbol{\alpha}$ values, and then selects as estimate the configuration yielding the highest marginal likelihood. As highlighted in Section 1, $\boldsymbol{\alpha}$ comprises few hyperparameters in routine GP implementations, and prediction is typically robust to minor variations in $\boldsymbol{\alpha}$, thereby making these grid strategies practically feasible and still reliable in common applications (e.g., Kuss and Rasmussen 2005; Rasmussen and Williams 2006; Nickisch and Rasmussen 2008; Riihimäki, Jylänki, and Vehtari 2013); see also the final discussion in Section 5 for additional details and possible solutions regarding the computational bottlenecks of the proposed grid search in situations when the number of hyperparameters in $\boldsymbol{\alpha}$ is moderate-to-large.

2.2. Evaluation via Functionals of Truncated Normals

The methodologies in Section 2.1 allow substantial improvements in terms of accuracy and scalability in the evaluation of predictive probabilities, but still require to deal with multivariate Gaussian cumulative distribution functions, a challenging task, especially in high dimensions. To overcome this issue, we derive an alternative expression for $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$ relying on functionals of multivariate truncated normals which are then approximated via mean-field variational Bayes (e.g., Blei, Kucukelbir, and McAuliffe 2017) to facilitate simple Monte Carlo evaluation of $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$ using samples from univariate truncated normals.

To derive this alternative expression, we shall first notice that the joint posterior $p(f(\mathbf{x}_{n+1}), \mathbf{f}(\mathbf{X}) \mid \mathbf{y})$ in (2) can be factorized as $p(f(\mathbf{x}_{n+1}) \mid \mathbf{f}(\mathbf{X}))p(\mathbf{f}(\mathbf{X}) \mid \mathbf{y})$, provided that $f(\mathbf{x}_{n+1})$ does not appear in the likelihood for $\mathbf{y}$, which is true because there is no overlap among predictors. Exploiting the well-known properties of GPs (Rasmussen and Williams 2006), the first factor in the above expression can be easily derived by applying the closure under conditioning property of multivariate Gaussians, thus, obtaining the univariate normal density

$$\begin{aligned} &p(f(\mathbf{x}_{n+1}) \mid \mathbf{f}(\mathbf{X})) \\ &\quad = \phi(f(\mathbf{x}_{n+1}) - (\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}}\mathbf{f}(\mathbf{X})); \sigma^2_{x_{n+1}}), \end{aligned} \tag{8}$$

with $\mu_{x_{n+1}} = m(\mathbf{x}_{n+1}) - \mathbf{H}_{x_{n+1}}\boldsymbol{\xi}$, $\mathbf{H}_{x_{n+1}} = \boldsymbol{\Omega}^*_{[n+1,1:n]}\boldsymbol{\Omega}^{-1}$ and $\sigma^2_{x_{n+1}} = K(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) - \boldsymbol{\Omega}^*_{[n+1,1:n]}\boldsymbol{\Omega}^{-1}\boldsymbol{\Omega}^*_{[1:n,n+1]}$, where the different quantities entering these expressions are defined as in (1) and (4). By adapting the recent conjugacy results for probit models with Gaussian priors in Durante (2019) to this GP setting, it is also possible to show that $p(\mathbf{f}(\mathbf{X}) \mid \mathbf{y})$ is the density of the unified skew-normal (SUN) (Arellano-Valle and Azzalini 2006) $\text{SUN}_{n,n}(\boldsymbol{\xi}, \boldsymbol{\Omega}, \bar{\boldsymbol{\Omega}}\boldsymbol{\omega}\mathbf{D}^\intercal\mathbf{s}^{-1}, \mathbf{s}^{-1}\mathbf{D}\boldsymbol{\xi}, \mathbf{s}^{-1}(\mathbf{D}\boldsymbol{\Omega}\mathbf{D}^\intercal + \mathbf{I}_n)\mathbf{s}^{-1})$, with $\mathbf{s} = [(\mathbf{D}\boldsymbol{\Omega}\mathbf{D}^\intercal + \mathbf{I}_n) \odot \mathbf{I}_n]^{1/2}$, $\bar{\boldsymbol{\Omega}} = \boldsymbol{\omega}^{-1}\boldsymbol{\Omega}\boldsymbol{\omega}^{-1}$ and $\boldsymbol{\omega} = (\boldsymbol{\Omega} \odot \mathbf{I}_n)^{1/2}$. Indeed, recalling the results in Sections 1–2.1 and applying the Bayes' rule, we have that $p(\mathbf{f}(\mathbf{X}) \mid \mathbf{y}) \propto p(\mathbf{f}(\mathbf{X}))p(\mathbf{y} \mid \mathbf{f}(\mathbf{X})) = \phi_n(\mathbf{f}(\mathbf{X}) - \boldsymbol{\xi}; \boldsymbol{\Omega})\Phi_n(\mathbf{D}\mathbf{f}(\mathbf{X}); \mathbf{I}_n)$, which is the kernel of a SUN density—as shown in the proof of Theorem 1 by Durante (2019). This class of random variables introduces asymmetric shapes within Gaussian densities via a skewness-inducing mechanism driven by the cumulative distribution function of an n-variate Gaussian with a full-rank covariance matrix. Hence, the evaluation of $p(\mathbf{f}(\mathbf{X}) \mid \mathbf{y})$ still requires calculation of multivariate Gaussian probabilities, leading to the same issues discussed in Section 2.1; see Arellano-Valle and Azzalini (2006), Azzalini and Capitanio (2014) and Durante (2019) for an in-depth discussion on the properties of SUN variables for posterior inference.

A possible option to address the above issue is to consider the discrete-choice interpretation of the probit GP introduced in Section 1. Under this representation, model (1) can be re-expressed as $y_i = 1(z_i > 0)$, where $(z_i \mid f(\mathbf{x}_i)) \sim \text{N}(f(\mathbf{x}_i), 1)$, independently for $i = 1, \ldots, n$, and $\mathbf{f}(\mathbf{X}) = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\intercal \sim \text{N}_n(\boldsymbol{\xi}, \boldsymbol{\Omega})$. Adapting the results in Holmes and Held (2006) to our GP setting, the joint posterior $p(\mathbf{f}(\mathbf{X}), \mathbf{z} \mid \mathbf{y})$ of $\mathbf{f}(\mathbf{X})$ and the augmented data $\mathbf{z} = (z_1, \ldots, z_n)^\intercal$, factorizes as $p(\mathbf{f}(\mathbf{X}) \mid \mathbf{z})p(\mathbf{z} \mid \mathbf{y})$, with

$$\begin{aligned} &p(\mathbf{f}(\mathbf{X}) \mid \mathbf{z}) \\ &\quad = \phi_n(\mathbf{f}(\mathbf{X}) - (\boldsymbol{\Omega}^{-1} + \mathbf{I}_n)^{-1}(\boldsymbol{\Omega}^{-1}\boldsymbol{\xi} + \mathbf{z}); (\boldsymbol{\Omega}^{-1} + \mathbf{I}_n)^{-1}) \\ &\qquad = \phi_n(\mathbf{f}(\mathbf{X}) - (\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{z}); \boldsymbol{\Sigma}_{\mathbf{X}}), \\ &p(\mathbf{z} \mid \mathbf{y}) \\ &\quad \propto \phi_n(\mathbf{z} - \boldsymbol{\xi}; \mathbf{I}_n + \boldsymbol{\Omega}) \textstyle\prod_{i=1}^n 1[(2y_i - 1)z_i > 0] \\ &\qquad = \phi_n(\mathbf{z} - \boldsymbol{\xi}; \boldsymbol{\Sigma}_{\mathbf{z}}) \textstyle\prod_{i=1}^n 1[(2y_i - 1)z_i > 0], \end{aligned} \tag{9}$$

where $\boldsymbol{\Sigma}_{\mathbf{X}} = (\boldsymbol{\Omega}^{-1} + \mathbf{I}_n)^{-1}$, $\boldsymbol{\mu}_{\mathbf{X}} = \boldsymbol{\Sigma}_{\mathbf{X}}\boldsymbol{\Omega}^{-1}\boldsymbol{\xi}$ and $\boldsymbol{\Sigma}_{\mathbf{z}} = \mathbf{I}_n + \boldsymbol{\Omega}$. Therefore, the joint posterior density $p(\mathbf{f}(\mathbf{X}) \mid \mathbf{z})p(\mathbf{z} \mid \mathbf{y})$ factorizes as the product of a Gaussian for $p(\mathbf{f}(\mathbf{X}) \mid \mathbf{z})$ and a multivariate truncated normal for $p(\mathbf{z} \mid \mathbf{y})$ obtained via component-wise truncation of $\text{N}_n(\boldsymbol{\xi}, \boldsymbol{\Sigma}_{\mathbf{z}})$ below or above 0, depending on whether $y_i = 1$ or $y_i = 0$, respectively, for $i = 1, \ldots, n$. As shown in Proposition 2, by combining Equations (8)–(9) with Lemma 7.1 in Azzalini and Capitanio (2014), it is possible to obtain an alternative expression for $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$ based on functionals of multivariate truncated normals. See the Appendix for a detailed proof.

Proposition 2. Under model (1), the predictive probability for a new response $y_{n+1} \in \{0; 1\}$ with predictor $\mathbf{x}_{n+1} \in \mathbb{R}^q$ is

$$\begin{aligned} &\text{pr}(y_{n+1} = 1 \mid \mathbf{y}) = 1 - \text{pr}(y_{n+1} = 0 \mid \mathbf{y}) \\ &= \mathbb{E}_{\mathbf{z}|\mathbf{y}}[\mathbb{E}_{\mathbf{f}(\mathbf{X})|\mathbf{z}}(\mathbb{E}_{f(\mathbf{x}_{n+1})|\mathbf{f}(\mathbf{X})}[\Phi(f(\mathbf{x}_{n+1}))])] \\ &= \mathbb{E}_{\mathbf{z}|\mathbf{y}}(\mathbb{E}_{\mathbf{f}(\mathbf{X})|\mathbf{z}}[\Phi(\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}}\mathbf{f}(\mathbf{X}); 1 + \sigma^2_{x_{n+1}})]) \qquad (10) \\ &= \mathbb{E}_{\mathbf{z}|\mathbf{y}}[\Phi(\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}}(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{z}); \\ &\qquad\qquad 1 + \sigma^2_{x_{n+1}} + \mathbf{H}_{x_{n+1}}\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{H}^{\intercal}_{x_{n+1}})], \end{aligned}$$

where the quantities in (10) are defined as in Equations (8) and (9), whereas $\mathbb{E}_{\mathbf{z}|\mathbf{y}}(\cdot)$ denotes the expectation with respect to the multivariate truncated normal density $p(\mathbf{z} \mid \mathbf{y})$ in (9).

Leveraging Proposition 2 it is possible to evaluate $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$ via Monte Carlo strategies based on independent samples from the multivariate truncated normal with density as in (9), thereby, producing the estimate $\hat{\text{pr}}(y_{n+1} = 1 \mid \mathbf{y}) = \sum_{r=1}^{R} \Phi(\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}}(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{z}^{(r)}); 1 + \sigma^2_{x_{n+1}} + \mathbf{H}_{x_{n+1}}\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{H}^{\intercal}_{x_{n+1}})/R$, where $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(R)}$ are independent and identically distributed samples from $p(\mathbf{z} \mid \mathbf{y})$ in (9). Unfortunately, sampling from multivariate truncated normals in settings where n is larger than a few hundreds raises the same computational issues discussed in Section 2.1, that is, the evaluation of multivariate Gaussian cumulative distribution functions (Holmes and Held 2006; Botev 2017; Pakman and Paninski 2014; Durante 2019; Fasano, Durante, and Zanella in press).

To avoid these issues, we adapt ideas in Fasano, Durante, and Zanella (in press) and propose to replace the intractable sampling density $p(\mathbf{z} \mid \mathbf{y})$ with a mean-field approximation $q^*(\mathbf{z}) = \prod_{i=1}^{n} q^*(z_i)$ factorizing over marginals $q^*(z_1), \ldots, q^*(z_n)$. In this way, the Monte Carlo estimate for $\text{pr}(y_{n+1} = 1 \mid \mathbf{y})$ can be obtained by sampling R times from n independent univariate approximate densities $q^*(z_1), \ldots, q^*(z_n)$ instead of the exact but intractable joint density $p(\mathbf{z} \mid \mathbf{y})$. Recalling the classical mean-field variational Bayes (VB) framework (e.g., Blei, Kucukelbir, and McAuliffe 2017), the optimal approximating density $q^*(\mathbf{z})$ is the one that minimizes the Kullback–Leibler (KL) divergence $\text{KL}[q(\mathbf{z}) \| p(\mathbf{z} \mid \mathbf{y})] = \mathbb{E}_{q(\mathbf{z})}(\log[q(\mathbf{z})/p(\mathbf{z} \mid \mathbf{y})])$ (Kullback and Leibler 1951) to $p(\mathbf{z} \mid \mathbf{y})$ among all the densities within the mean-field family $\mathcal{Q} = \{q(\mathbf{z}) : q(\mathbf{z}) = \prod_{i=1}^{n} q(z_i)\}$. The solution of such a minimization problem is, typically, not available in closed form but can be obtained via coordinate ascent variational inference (CAVI) algorithms (Bishop 2006; Blei, Kucukelbir, and McAuliffe 2017) that iteratively minimize the KL with respect to each component $q(z_i)$ at a time, keeping fixed the others at their most recent estimate $\mathbf{q}^{(t-1)}(\mathbf{z}_{-i})$, where $\mathbf{z}_{-i}$ denotes vector $\mathbf{z}$ without the i-th entry. Recalling Bishop (2006), this is accomplished via the updates

$$q^{(t)}(z_i) \propto \exp[\mathbb{E}_{\mathbf{q}^{(t-1)}(\mathbf{z}_{-i})}(\log[p(z_i \mid \mathbf{z}_{-i}, \mathbf{y})])], \qquad (11)$$

for each $i = 1, \ldots, n$, at iteration t, until convergence. In (11), the quantity $p(z_i \mid \mathbf{z}_{-i}, \mathbf{y})$ denotes the full conditional density of z_i. Due to the closure under conditioning property of the multivariate truncated normal (Horrace 2005), such a quantity can be derived explicitly from $p(\mathbf{z} \mid \mathbf{y})$ in (9) and coincides with the density of a univariate truncated normal. In particular, we can express each $p(z_i \mid \mathbf{z}_{-i}, \mathbf{y})$ as

$$\begin{aligned} &p(z_i \mid \mathbf{z}_{-i}, \mathbf{y}) \qquad (12) \\ &\propto \phi(z_i - [\xi_i + \mathbf{H}_{z_i}(\mathbf{z}_{-i} - \boldsymbol{\xi}_{-i})]; \sigma^2_{z_i}) 1[(2y_i - 1)z_i > 0], \end{aligned}$$

where $\boldsymbol{\xi}_{-i}$ denotes the prior mean vector $\boldsymbol{\xi}$ without the i-th element, whereas $\mathbf{H}_{z_i} = \boldsymbol{\Sigma}_{\mathbf{z}[i,-i]}(\boldsymbol{\Sigma}_{\mathbf{z}[-i,-i]})^{-1}$ and $\sigma^2_{z_i} = \boldsymbol{\Sigma}_{\mathbf{z}[i,i]} - \boldsymbol{\Sigma}_{\mathbf{z}[i,-i]}(\boldsymbol{\Sigma}_{\mathbf{z}[-i,-i]})^{-1}\boldsymbol{\Sigma}_{\mathbf{z}[-i,i]}$. Density in (12) has a log-kernel which is linear in $\mathbf{z}_{-i}$ and, therefore, replacing the expression for $p(z_i \mid \mathbf{z}_{-i}, \mathbf{y})$ within the CAVI updates in Equation (11), it follows that also $q^{(t)}(z_i)$ has a univariate truncated normal density as in (12) with $\mathbf{z}_{-i}$ replaced by

$$\begin{aligned} \mathbf{z}^{(t-1)}_{-i} = [&\mathbb{E}_{q^{(t)}(z_1)}(z_1), \ldots, \mathbb{E}_{q^{(t)}(z_{i-1})}(z_{i-1}), \\ &\mathbb{E}_{q^{(t-1)}(z_{i+1})}(z_{i+1}), \ldots, \mathbb{E}_{q^{(t-1)}(z_n)}(z_n)]^{\intercal}. \end{aligned}$$

Each term in $\mathbf{z}^{(t-1)}_{-i}$ is the expectation of a univariate truncated normal, that is explicitly available, thus, producing a simple CAVI relying on closed-form updates; see Algorithm 2.

Once the optimal univariate truncated normal approximating densities $q^*(z_1), \ldots, q^*(z_n)$ are available, Equation (10) can be easily evaluated via Monte Carlo by letting

$$\begin{aligned} &\hat{\text{pr}}(y_{n+1} = 1 \mid \mathbf{y}) \\ &= \frac{1}{R}\sum_{r=1}^{R} \Phi(\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}}(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{z}^{*(r)}); \qquad (13) \\ &\qquad\qquad 1 + \sigma^2_{x_{n+1}} + \mathbf{H}_{x_{n+1}}\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{H}^{\intercal}_{x_{n+1}}), \end{aligned}$$

Algorithm 2: Compute (10) via Monte Carlo as in (13) based on the mean-field approximation of $p(\mathbf{z} \mid \mathbf{y})$

CAVI algorithm

[a] Precompute $\boldsymbol{\Omega}^{-1}$ and $\boldsymbol{\Sigma}^{-1}_{\mathbf{z}} = (\mathbf{I}_n + \boldsymbol{\Omega})^{-1}$, and leverage the standard properties for the inverse of block matrices to obtain $\mathbf{H}_{z_i}$ and $\sigma^2_{z_i}$, for each $i = 1, \ldots, n$ as suitable sub-blocks of $\boldsymbol{\Sigma}^{-1}_{\mathbf{z}}$.

[b] Initialize $\mathbf{z}^{(0)} \in \mathbb{R}^n$, and apply CAVI to obtain the optimal mean-field approximation $q^*(\mathbf{z}) = \prod_{i=1}^{n} q^*(z_i)$ for $p(\mathbf{z} \mid \mathbf{y})$.

for $t = 1$ *until convergence* **do**

  for $i = 1, \ldots, n$ **do**

    Set the approximating density for z_i at step t equal to $q^{(t)}(z_i) \propto \phi(z_i - [\xi_i + \mathbf{H}_{z_i}(\mathbf{z}^{(t-1)}_{-i} - \boldsymbol{\xi}_{-i})]; \sigma^2_{z_i}) 1[(2y_i - 1)z_i > 0]$ with $\mathbf{z}^{(t-1)}_{-i} = [\mathbb{E}_{q^{(t)}(z_1)}(z_1), \ldots, \mathbb{E}_{q^{(t)}(z_{i-1})}(z_{i-1}), \mathbb{E}_{q^{(t-1)}(z_{i+1})}(z_{i+1}), \ldots, \mathbb{E}_{q^{(t-1)}(z_n)}(z_n)]^{\intercal}$.

  Output: $q^*(\mathbf{z}) = \prod_{i=1}^{n} q^*(z_i)$, where each $q^*(z_i)$ is a univariate truncated normal.

Evaluation of predictive probabilities

[c] Compute $\boldsymbol{\Omega}^{-1}\boldsymbol{\Sigma}_{\mathbf{X}}$ which enters the definition of the key quantities in (13), namely $\mathbf{H}_{x_{n+1}}\boldsymbol{\mu}_{\mathbf{X}}$ and $\mathbf{H}_{x_{n+1}}\boldsymbol{\Sigma}_{\mathbf{X}}$. Note that, by standard properties of matrix inverse $\boldsymbol{\Omega}^{-1}\boldsymbol{\Sigma}_{\mathbf{X}} = \boldsymbol{\Omega}^{-1}(\boldsymbol{\Omega}^{-1} + \mathbf{I}_n)^{-1} = (\mathbf{I}_n + \boldsymbol{\Omega})^{-1}$, which coincides with $\boldsymbol{\Sigma}^{-1}_{\mathbf{z}}$ already precomputed in **[a]**.

[d] Estimate (10) via Monte Carlo as in (13), based on R independent samples from the optimal univariate truncated normal approximating densities provided by step **[b]**.

with $\mathbf{z}^{*(r)} = (z_1^{*(r)}, \ldots, z_n^{*(r)})^\intercal$, where each $z_i^{*(r)}$ can be efficiently sampled from the corresponding univariate truncated normal approximating density $q^*(z_i)$, independently for $i = 1, \ldots, n$ and $r = 1, \ldots, R$. Unlike for the multivariate case, sampling from univariate truncated normals can be effectively done in standard statistical softwares, thus, avoiding issues in large n settings.

Algorithm 2 provides the pseudo-code to implement the proposed VB approximation for the predictive probabilities in (10). As is clear from Algorithm 2, the quantities $\mathbf{H}_{z_i}$ and $\sigma^2_{z_i}$, $i = 1, \ldots, n$, involved in step **[b]**, coincide with suitable sub-blocks of $\boldsymbol{\Sigma}_{\mathbf{z}}^{-1}$. Due to this, the operations required to update each $q^{(t)}(z_i)$ in **[b]** are linear in n, and, therefore, the overall cost of CAVI is $\mathcal{O}(n^3)$, which coincides with the cost for precomputing matrix $\boldsymbol{\Sigma}_{\mathbf{z}}^{-1}$ in **[a]**. Leveraging these results, the evaluation of the predictive probabilities in step **[d]** implies an $\mathcal{O}(n)$ cost per Monte Carlo sample, since, according to step **[c]**, the main quantities in (13) can be derived from those precomputed in **[a]**. This yields a total cost for Algorithm 2 of $\mathcal{O}(n^3 + Rn)$ which reduces by $n^{1/2}$ the Monte Carlo complexity of Algorithm 1, but increases by the same amount the precomputation cost. As for Algorithm 1, also in Algorithm 2 the most computationally intensive steps in **[a]**–**[c]** do not need to be re-executed each time a new prediction is required, thereby making computation of predictive probabilities at multiple data points almost as expensive as implementing this task for a single location.

As discussed, for example, in Kuss and Rasmussen (2005), Riihimäki, Jylänki, and Vehtari (2013), the cubic cost is commonly unavoidable in standard GP settings with generic covariance matrix. However, unlike for alternative approximations relying, for instance, on expectation-propagation (EP) methods (e.g., Kuss and Rasmussen 2005; Riihimäki, Jylänki, and Vehtari 2013), this $\mathcal{O}(n^3)$ cost is only paid once in the precomputation step, and not for each iteration of the optimization routine. This yields substantial improvements in terms of scalability to high dimensions relative to EP. As outlined in the simulation studies in Section 3, these gains are obtained without sacrificing estimation accuracy, when compared to nonapproximate methods. This is due to the fact that the proposed strategy integrates out $\mathbf{f}(\mathbf{X})$ analytically in (10) with respect to its exact density $p(\mathbf{f}(\mathbf{X}) \mid \mathbf{z})$, and only approximates $p(\mathbf{z} \mid \mathbf{y})$. This departs from classical VB solutions (Girolami and Rogers 2006) which consider a mean-field approximation $q^*(\mathbf{f}(\mathbf{X})) \prod_{i=1}^n q^*(z_i)$ of the joint density $p(\mathbf{f}(\mathbf{X}), \mathbf{z} \mid \mathbf{y})$, and then compute predictive probabilities based on Monte Carlo samples from $q^*(\mathbf{f}(\mathbf{X}))$. This yields less accurate estimates of the predictive probabilities that, unlike for the solution we propose, do not fully incorporate the exact dependence between $\mathbf{f}(\mathbf{X})$ and $\mathbf{z}$ (e.g., Nickisch and Rasmussen 2008, Figure 6; Fasano, Durante, and Zanella in press).

3. Simulation Studies

In this section, we study the gains in accuracy and computational scalability of the methods developed in Sections 2.1 and 2.2 relative to state-of-the-art alternatives. More specifically, to quantify the magnitude of the improvements provided by the tile-low-rank (TLR) strategy developed in Section 2.1, we consider as a competitor the recent minimax tilting method (TN) by Botev (2017) (see R package `TruncatedNormal`), which is used here to evaluate the Gaussian cumulative distribution functions involved in the predictive probability (4). This strategy has been shown to substantially improve the accuracy and computational tractability of other state-of-the-art solutions and, hence, provides a challenging benchmark to assess the gains of the TLR procedure. The performance improvements of the VB developed in Section 2.2 are, instead, compared against Monte Carlo inference under the widely-used STAN implementation of the Hamiltonian no-u-turn sampler (Hoffman and Gelman 2014) available in the state-of-the-art R package `rstan`. Both VB and STAN provide Monte Carlo estimates of predictive probabilities but, unlike for our proposed VB solution, STAN relies on samples from the exact posterior, thus, providing a relevant and routinely used competitor for evaluating the accuracy of the proposed VB approximation and its gains in runtime. As discussed in Section 2.2, classical mean-field variational methods (e.g., Girolami and Rogers 2006) and EP solutions (e.g., Kuss and Rasmussen 2005; Riihimäki, Jylänki, and Vehtari 2013) would yield reduced accuracy or higher computational costs than the proposed VB, and, hence, are not implemented.

To evaluate the performance in high-dimensional settings, we generate the binary response data on the 100×100 unit grid $\mathcal{G} = \{\mathbf{x} = (x_1, x_2) : x_1 = (1/100, 2/100, \ldots, 100/100), x_2 = (1/100, 2/100, \ldots, 100/100)\}$ with equally-spaced predictors, thereby obtaining $n = 10{,}000$ nonoverlapping configurations. At these locations, we simulate $y_1, \ldots, y_{10{,}000}$ from independent Bernoullis with probabilities $\Phi(f_0(\mathbf{x}_1)), \ldots, \Phi(f_0(\mathbf{x}_{10{,}000}))$ displayed in Figure 2, where $\mathbf{f}_0(\mathbf{X}) = [f_0(\mathbf{x}_1), \ldots, f_0(\mathbf{x}_{10{,}000})]^\intercal$ is a sample from a GP having mean $m(\mathbf{x}) = 0$ and squared exponential covariance kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp(-[\alpha_1^2(x_1 - x_1')^2 + \alpha_2^2(x_2 - x_2')^2]),$$

with $\boldsymbol{\alpha} = (\alpha_1, \alpha_2) = (\sqrt{30}, \sqrt{30})$ to illustrate also performance in estimating more than one GP hyperparameter; see also Section 5 for a discussion on hyperparameter estimation in higher dimensional settings. The proportion of '1's and '0's in the 10,000 simulated binary responses is 49.5% and 50.5%, respectively, thus, providing a balanced dataset. To assess performance in estimating the predictive probabilities, we adopt a validation-set approach by simulating probability parameters and the associated binary responses for 100 out-of-sample units under two scenarios. As outlined in Figure 2, the first one relies on randomly distributed locations, whereas the second focuses on a grid structure, and both comprise relatively balanced binary responses, as for the training sample. To provide a more comprehensive assessment, we also compare performance in lower-dimensional training problems with $n \in \{15^2; 25^2; 50^2\}$ obtained by selecting a $n^{1/2} \times n^{1/2}$ sub-grid of $\mathcal{G}$ with equally-spaced configurations between 0 and 1, along with their associated probability parameters and simulated responses.

Table 1 summarizes the accuracy and computational scalability of the methods analyzed, at varying n and under the two different scenarios considered for prediction. In reporting the results, we set conservative computational budget of one day and compute the out-of-sample validation MSEs instead of the cross-validated ones to limit the overall computational effort within our capacity, especially for the two competitors TN and

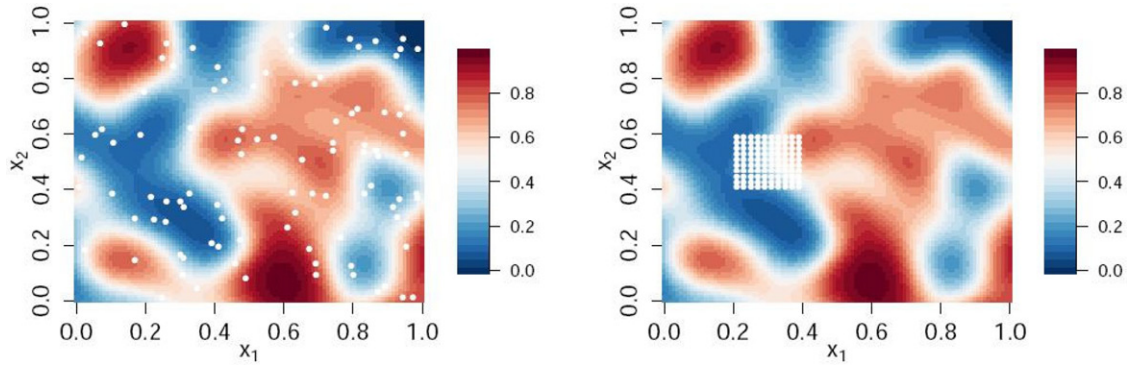

Figure 2. Simulated probabilities on the 100 × 100 grid $\mathcal{G} = \{\mathbf{x} = (x_1, x_2) : x_1 = (1/100, 2/100, \ldots, 100/100), x_2 = (1/100, 2/100, \ldots, 100/100)\}$ in the unit square, where $f(\mathbf{x})$ is a zero mean GP with squared exponential covariance kernel. White circles denote the 100 test locations distributed randomly (left) and on a grid (right), used for prediction.

Table 1. Runtimes and accuracy in estimating out-of-sample predictive probabilities, at varying training sample size n, of STAN (Hoffman and Gelman 2014), TN (Botev 2017), TLR (Section 2.1) and VB (Section 2.2), when the 100 test locations are distributed either randomly [random] or on a grid [grid].

Method	Performance measures	$n = 225$	$n = 625$	$n = 2500$	$n = 10{,}000$
STAN	TIME [seconds]	1382	18,066	—	—
	MSE [random]	0.015	0.014	—	—
	MSE [grid]	0.023	0.015	—	—
TN	TIME [seconds]	7	41	—	—
	MSE [random]	0.017	0.014	—	—
	MSE [grid]	0.027	0.017	—	—
TLR	TIME [seconds]	1	5	37	250
	MSE [random]	0.017	0.014	0.005	0.002
	MSE [grid]	0.025	0.019	0.007	0.003
VB	TIME [seconds]	1	3	23	898
	MSE [random]	0.016	0.014	0.005	0.001
	MSE [grid]	0.025	0.017	0.004	0.001

NOTE: TIME: runtime in seconds for predicting at one location. MSE: mean squared error between the 100 estimated predictive probabilities and the true ones. Empty cells refer to situations in which the overall runtime of the whole prediction task exceeded the conservative budget of one day.

STAN. To provide a reliable comparison between the different implementations, we consider the runtime for predicting one test unit. Such a measure complements the formal computational complexities derived in detail in Sections 2.1–2.2, and comprises also the precomputation costs, which, however, do not need to be paid once again when predicting at multiple locations. For instance, in our implementation of the VB strategy in *https://github.com/danieledurante/PredProbitGP*, the overall runtime in seconds for predicting at 100 locations almost coincides with the one reported in Table 1 for a single prediction.

As illustrated in the tutorial implementations of all the methods analyzed—which are available at *https://github.com/danieledurante/PredProbitGP/blob/main/Tutorial.md*—Monte Carlo inference under STAN (Hoffman and Gelman 2014) relies on the highly-optimized state-of-the-art `R` package `rstan` applied to model (1) for obtaining posterior samples from $\mathbf{f}(\mathbf{X})$, which are then used to compute the predictive probabilities at the test locations via ordinary kriging. Such evaluations rely on 10,000 MCMC samples after a burnin of 10,000, setting the true $\boldsymbol{\alpha} = (\sqrt{30}, \sqrt{30})$. In evaluating the performance of minimax tilting (TN) (Botev 2017), we compute the numerator and the denominator in (4) separately via the `R` package `TruncatedNormal`, using the default settings. Equation (4) is also evaluated under the TLR method presented in Section 2.1 and summarized in Algorithm 1, which can be implemented via simple adaptations of the R package `tlrmvnmvt` (Cao et al. 2021). In implementing this routine, we set the block size to $n^{1/2}$, the truncation level to 10^{-4} and $R = 20{,}000$. To evaluate the predictive probabilities under TN and TLR, we avoid setting $\boldsymbol{\alpha}$ at the true values $(\sqrt{30}, \sqrt{30})$, but instead estimate these two GP hyperparameters via the grid search discussed in Section 2.1, that evaluates the marginal likelihood in (3) on a 10×10 grid in $[\sqrt{15}, \sqrt{45}] \times [\sqrt{15}, \sqrt{45}] \in \mathbb{R}^2$ leveraging the `R` packages `TruncatedNormal` and `tlrmvnmvt`, for TN and TLR, respectively. The results are comparable, although TLR requires substantially lower runtimes. The estimate of $\boldsymbol{\alpha}$ provided by `tlrmvnmvt` is also used in the implementation of the VB presented in Section 2.2 and summarized in Algorithm 2. Also in this context we consider $R = 20{,}000$ Monte Carlo samples to evaluate (10) via (13). Such values are generated from the optimal univariate truncated normal approximating densities produced by the CAVI in Algorithm 2, which can be implemented via minor adaptations of the code in the GitHub repository `Probit-PFMVB` (Fasano, Durante, and Zanella in press).

As clarified in Table 1, the methods proposed in Sections 2.1 and 2.2 notably reduce the runtimes relative to state-of-the-art competitors, thus, making prediction under probit GP computationally feasible in those high-dimensional settings that often arise in various applications. According to the MSEs reported in Table 1, such a notable reduction in runtimes under TLR and VB is crucially obtained at almost no costs in terms of accuracy in the estimation of the predictive probabilities, when compared to relevant competitors relying on MCMC samples from the exact posterior (STAN) or on accurate evaluation of multivariate Gaussian cumulative distribution functions (TN). The runtimes of TLR and VB are also coherent with the associated $\mathcal{O}(n^{5/2} + Rn^{3/2})$ and $\mathcal{O}(n^3 + Rn)$ computational costs discussed in Sections 2.1–2.2, which make VB more competitive in small-to-moderate dimensions, and TLR more suitable in much higher dimensions due to the reduction of the cubic precomputation cost. All computations were run on a 3.4 GHz Intel Core i5 CPU workstation, without multithreading.

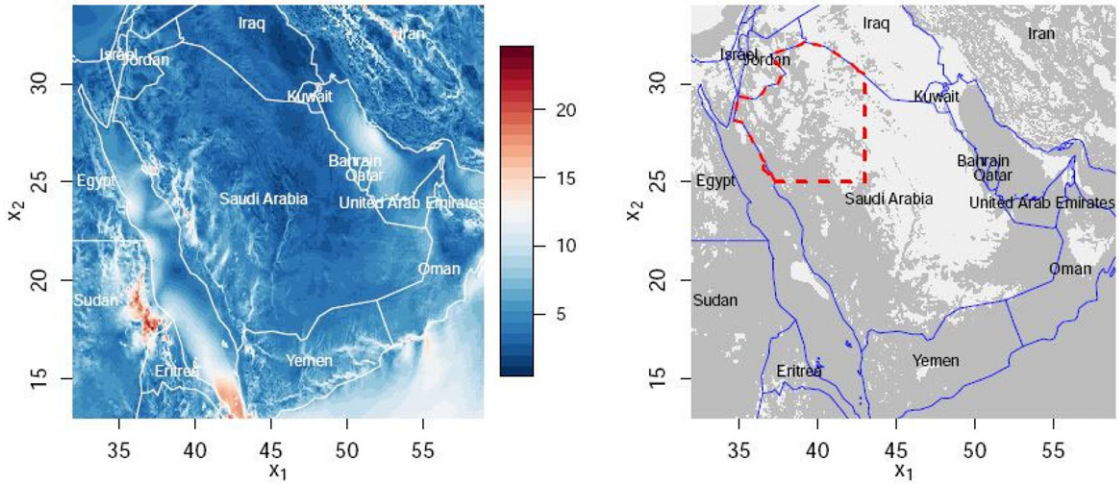

Figure 3. Heatmaps representing the windspeed at 140 m high (left) and a binary version y of this measure defining whether the local windspeed is sufficiently high for energy production (dark gray: YES; light gray: NO) based on the 4 m/s threshold (right) on January 21, 2014. The dashed area denotes the spatial region that is used for modeling and prediction.

4. Saudi Arabia Windspeed Application

We conclude by applying the methods developed in Sections 2.1 and 2.2 to a real-world environmental application aimed at modeling whether the local windspeed exceeds a prespecified working threshold for energy production in a given region of interest in Saudi Arabia. Wind turbines for generating electricity typically have two windspeed thresholds, of which the lower controls when the blades of the turbine start to be in motion and the higher indicates if the turbine should be switched off to avoid strong-wind damage. Here, the binary response $y_i \in \{0; 1\}$ measures whether the windspeed at the i-th location exceeds the lower threshold, thus, allowing production of wind power, which is referred to as the working threshold of wind turbines. This important application is motivated by the growing domestic energy consumption in Saudi Arabia and by the attempt to reduce the reliance on fossil fuels, thereby leading to an increasing interest on renewable energy sources, including wind (Shaahid, Al-Hadhrami, and Rahman 2014; Chen et al. 2018; Tagle et al. 2019; Giani et al. 2020). The effective exploitation of such resources and the careful management of the energy stations require careful modeling and prediction at a fine spatial resolution of whether the local windspeed exceeds or not a given threshold for energy production. As discussed in the following, this fine grid of observations commonly produces a sample size around tens of thousands units. This makes state-of-the-art algorithms for probit GP computationally unfeasible, thus, motivating the use of our scalable solutions in Sections 2.1–2.2.

The windspeed dataset considered in this article is produced by the Weather Research and Forecasting (WRF) model (Yip 2018), which constructs the weather system through partial differential equations on the mesoscale and demands strong computation capacity to serve meteorological applications (Skamarock et al. 2008). The time resolution of our data is daily and we use windspeed over the region of north-west Saudi Arabia on January 21, 2014 for modeling and out-of-sample prediction. Such a region covers the wind farm at Dumat Al Jandal, which is the first wind farm in Saudi Arabia and currently under construction, as well as the future smart city of NEOM, a strategic component of the Saudi 2030 Vision, where wind power is expected to be a key energy resource. Moreover, the windspeed on January 21, 2014 has high variability across this region, which makes the out-of-sample prediction task much more challenging. As shown in Figures 3 and 4 the region under analysis is obtained by intersecting the Saudi Arabia territorial map with the rectangle ranging from E34° 30′ to E43° and from N25° to N32°. Within this region we consider a fine grid of $n = 9036$ equally-spaced locations $\mathbf{x}_i = (x_{i1}, x_{i2})^\intercal = (\texttt{long}_i, \texttt{lat}_i)^\intercal$ at which we monitor whether the windspeed is either above ($y_i = 1$) or below ($y_i = 0$) the working threshold of wind turbines for each $i = 1, \ldots, 9036$. Following Chen et al. (2018), such a threshold is set at 4 m/s, leading to a balanced dataset with 51% '1' responses, and 49% observed '0's. Similar to Section 3, we monitor predictive performance at 100 out-of-sample locations displayed in Figure 4, which are distributed randomly, and on a grid centered at the Dumat Al Jandal wind farm.

Motivated by the results in the simulation study in Section 3, we consider a probit GP with zero mean and squared exponential covariance kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp(-[\alpha_1^2(x_1 - x_1')^2 + \alpha_2^2(x_2 - x_2')^2]),$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ is estimated via a grid maximization of the marginal likelihood in (3) evaluated via the `tlrmvnmvt` package on a 20×20 grid of values in $[1, \sqrt{30}] \times [1, \sqrt{30}]$. The estimated $\boldsymbol{\alpha}$ is $(3.59, 4.77)$, which interestingly implies a similarly rapid decay in correlation across the two spatial directions. This result is consistent with the abrupt changes of the binary responses. Recalling the results in Table 1, calculation of the predictive probabilities is only performed under the methods presented in Sections 2.1 (TLR) and 2.2 (VB) since STAN and TN would be computationally impractical in such a high-dimensional setting with $n = 9036$. Although this issue could be circumvented via subsampling, such a procedure is suboptimal since it reduces the sample size n and, as a consequence, it yields less accurate estimates of the predictive probabilities with higher MSE; see also Table 1. In implementing both methods, we set $\boldsymbol{\alpha} = (3.59, 4.77)$ and consider the same settings as in the

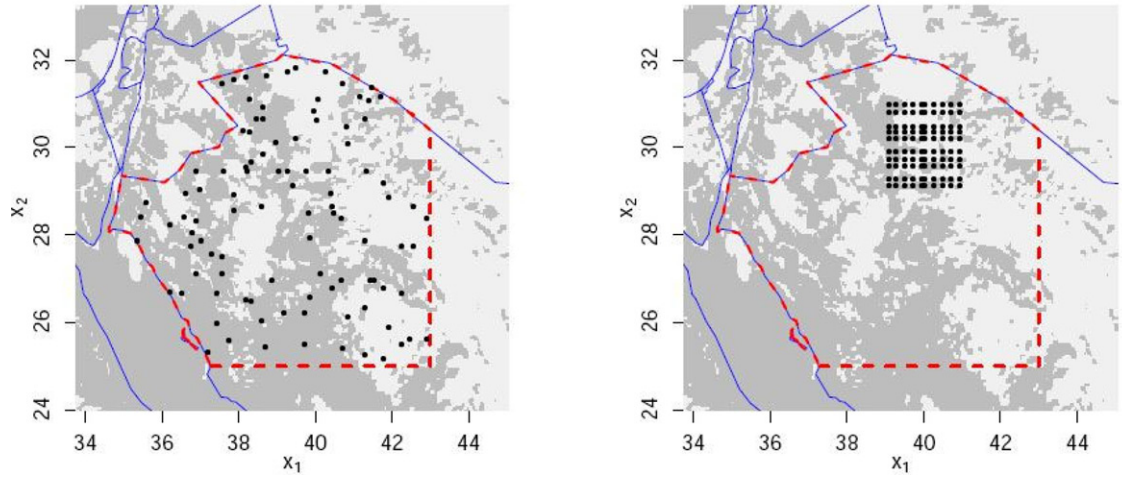

Figure 4. For the spatial region used in modeling and prediction, heatmaps defining whether the local windspeed is sufficiently high for energy production (dark gray: YES; light gray: NO) based on the 4 m/s threshold on January 21, 2014. Black circles denote the 100 test locations distributed randomly (left) and on a grid (right), used for prediction.

simulation study in Section 3, thus, obtaining runtimes that are comparable to those discussed in Section 3 for the scenario with $n = 10{,}000$. Out-of-sample predictive performance measured via the area under the ROC curve (AUC) is similarly accurate for both methods. In particular, the AUCs for the random and grid test scenarios are above 0.9 under both TLR and VB. This confirms the accuracy gains that can be obtained by the development of increasingly scalable strategies which can be effectively applied to larger samples sizes.

5. Discussion

This article provides novel expressions for the predictive probabilities under probit models with GP priors, relying either on multivariate Gaussian cumulative distribution functions or on functionals of multivariate truncated normals, and proposes scalable computational strategies to evaluate such quantities in common high-dimensional settings, thus, covering an important gap in the literature. As highlighted in the simulations studies in Section 3, such computational gains are notable and do not sacrifice accuracy. This allows effective exploitation of the full information in the observed data to improve predictive accuracy, even in computationally challenging applications, such as the windspeed study in Section 4, where the high sample size affects the practical feasibility of available state-of-the-art solutions.

The above results open up several avenues for future research. A relevant direction is to address the possible computational bottlenecks of the proposed grid search for hyperparameter tuning in settings when the dimension of $\boldsymbol{\alpha}$ is large. This issue arises in high-dimensional predictor domains when considering, for example, automatic relevance determination (ARD) kernels that assign a different scaling hyperparameter for each predictor (e.g., Rasmussen and Williams 2006, chap. 4.2 and 5.1). Direct application of the proposed grid search would be computationally challenging in this high-dimensional hyperparameter space as it would require an excessive number of evaluations of the marginal likelihood, unless some assumptions are made on the kernel function to reduce the number of hyperparameters. Although these simplifications are sometimes made in practice (e.g., Kuss and Rasmussen 2005; Nickisch and Rasmussen 2008), it would be still desirable to develop scalable tuning strategies in high-dimensional hyperparameter spaces. A promising direction to address this goal is to combine our improved strategy for the evaluation of the marginal likelihood in Section 2.1 with state-of-the-art machine learning algorithms for high-dimensional hyperparameter tuning that require a low number of evaluations of the objective function (e.g., Bergstra et al. 2011; Snoek, Larochelle, and Adams 2012; Klein et al. 2017).

Another area of interest is direct estimation and uncertainty quantification on linear relationships among the response and predictors, when included within the GP mean function via $\mathbf{x}^\intercal\boldsymbol{\beta}$. Although such a goal departs from the predictive focus of this article, it shall be noticed that the posterior distribution $p(\boldsymbol{\beta}|\mathbf{y})$ of the regression coefficients can be derived in closed form when considering Gaussian priors for $\boldsymbol{\beta}$. In particular, note that when $f(\mathbf{x})$ is a GP with mean function $m(\mathbf{x}) = \mathbf{x}^\intercal\boldsymbol{\beta}$ and covariance kernel $K(\mathbf{x}, \mathbf{x}')$, then, leveraging standard GP properties, it holds that

$$\mathbf{f}(\mathbf{X}) = \big[f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\big]^\intercal = \mathbf{X}\boldsymbol{\beta} + \bar{\mathbf{f}}(\mathbf{X}) = \bar{\mathbf{X}}\boldsymbol{\eta},$$

where $\bar{\mathbf{X}} = (\mathbf{X}, \mathbf{I}_n)$, $\boldsymbol{\eta} = [\boldsymbol{\beta}^\intercal, \bar{\mathbf{f}}(\mathbf{X})^\intercal]^\intercal$, and $\bar{\mathbf{f}}(\mathbf{X}) \sim \mathrm{N}_n(\mathbf{0}, \boldsymbol{\Omega})$, with $\boldsymbol{\Omega}$ defined as in (1). Hence, letting $\boldsymbol{\beta} \sim \mathrm{N}_q(\mathbf{0}, \delta^2\mathbf{I}_q)$, as in Section 1, it follows that $\boldsymbol{\eta} \sim \mathrm{N}_{q+n}(\mathbf{0}, \boldsymbol{\Omega}_{\boldsymbol{\eta}})$, where $\boldsymbol{\Omega}_{\boldsymbol{\eta}}$ is a $(q+n) \times (q+n)$ block-diagonal covariance matrix with blocks $\boldsymbol{\Omega}_{\boldsymbol{\eta}[1,1]} = \delta^2\mathbf{I}_q$ and $\boldsymbol{\Omega}_{\boldsymbol{\eta}[2,2]} = \boldsymbol{\Omega}$. Recalling Sections 1 and 2, this multivariate Gaussian prior, when combined with the probit likelihood via the Bayes' rule, yields the posterior distribution

$$p(\boldsymbol{\eta} \mid \mathbf{y}) \propto p(\boldsymbol{\eta})p(\mathbf{y} \mid \boldsymbol{\eta}) = \phi_{q+n}(\boldsymbol{\eta}; \boldsymbol{\Omega}_{\boldsymbol{\eta}})\Phi_n(\mathbf{D}_{\boldsymbol{\eta}}\boldsymbol{\eta}; \mathbf{I}_n),$$

with $\mathbf{D}_{\boldsymbol{\eta}} = \mathrm{diag}(2y_1 - 1, \ldots, 2y_n - 1)\bar{\mathbf{X}}$, whose kernel can be shown to coincide with that of the unified skew-normal variable

$$\mathrm{SUN}_{q+n,n}(\mathbf{0}, \boldsymbol{\Omega}_{\boldsymbol{\eta}}, \bar{\boldsymbol{\Omega}}_{\boldsymbol{\eta}}\boldsymbol{\omega}_{\boldsymbol{\eta}}\mathbf{D}_{\boldsymbol{\eta}}^\intercal\mathbf{s}_{\boldsymbol{\eta}}^{-1}, \mathbf{0}, \mathbf{s}_{\boldsymbol{\eta}}^{-1}(\mathbf{D}_{\boldsymbol{\eta}}\boldsymbol{\Omega}_{\boldsymbol{\eta}}\mathbf{D}_{\boldsymbol{\eta}}^\intercal + \mathbf{I}_n)\mathbf{s}_{\boldsymbol{\eta}}^{-1}),$$

with $\mathbf{s}_{\boldsymbol{\eta}} = [(\mathbf{D}_{\boldsymbol{\eta}}\boldsymbol{\Omega}_{\boldsymbol{\eta}}\mathbf{D}_{\boldsymbol{\eta}}^\intercal + \mathbf{I}_n) \odot \mathbf{I}_n]^{1/2}$, $\bar{\boldsymbol{\Omega}}_{\boldsymbol{\eta}} = \boldsymbol{\omega}_{\boldsymbol{\eta}}^{-1}\boldsymbol{\Omega}_{\boldsymbol{\eta}}\boldsymbol{\omega}_{\boldsymbol{\eta}}^{-1}$ and $\boldsymbol{\omega}_{\boldsymbol{\eta}} = (\boldsymbol{\Omega}_{\boldsymbol{\eta}} \odot \mathbf{I}_{q+n})^{1/2}$, leveraging the recent conjugacy results

in Theorem 1 of Durante (2019). Notably, such a class of distributions is closed under marginalization (Arellano-Valle and Azzalini 2006; Azzalini and Capitanio 2014), meaning that also the posterior distribution $p(\boldsymbol{\beta} \mid \mathbf{y})$ for $\boldsymbol{\beta}$—which corresponds to the first q entries in $\boldsymbol{\eta}$—is unified skew-normal with parameters that can be directly obtained from those of the joint SUN posterior for $\boldsymbol{\eta}$ via simple linear algebra operations; see Azzalini and Capitanio (2014, chap. 7.1.2) for details. This result facilitates estimation and uncertainty quantification for $\boldsymbol{\beta}$, when this is of interest, leveraging the functionals of the associated closed-form SUN posterior (Durante 2019).

Finally, it is worth emphasizing that the methods developed in Section 2 can be naturally adapted to any probit model with a multivariate Gaussian prior for the linear predictor. Relevant examples include classical Bayesian probit regression, multivariate probit models (e.g., Chib and Greenberg 1998; Fasano et al. 2021) and general additive representations relying on basis expansions. Extensions to categorical response data under a multinomial probit GP model or to more general SUN priors can also be explored by leveraging results in Durante (2019), Fasano and Durante (in press) and Benavoli, Azzimonti, and Piga (2020).

Appendix: Proof of Theoretical Results

To prove Propositions 1–2 let us first state the following Lemma.

Lemma 1 (Lemma 7.1 in Azzalini and Capitanio (2014)). If $\mathbf{U} \sim \mathrm{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$ then $\mathbb{E}[\Phi_q(\mathbf{H}^\intercal \mathbf{U} + \mathbf{k}; \boldsymbol{\Psi})] = \Phi_q(\mathbf{k}; \boldsymbol{\Psi} + \mathbf{H}^\intercal \boldsymbol{\Sigma} \mathbf{H})$, for any choice of the vector $\mathbf{k} \in \mathbb{R}^q$, the $p \times q$ matrix $\mathbf{H}$ and the $q \times q$ symmetric positive–definite matrix $\boldsymbol{\Psi}$.

Combining the closure under conditioning property of multivariate Gaussians with the above result—whose proof can be found in Azzalini and Capitanio (2014)—the proof of Propositions 1–2 can be obtained via simple derivations described below.

Proof of Proposition 1. To prove Proposition 1, first to notice that by application of the Bayes' rule

$$\mathrm{pr}(y_{n+1} = 1 \mid \mathbf{y}) = p(y_{n+1} = 1, \mathbf{y})/p(\mathbf{y}).$$

Hence, it suffices to show that

$$p(y_{n+1} = 1, \mathbf{y}) = \Phi_{n+1}(\mathbf{D}^* \boldsymbol{\xi}^*; \mathbf{I}_{n+1} + \mathbf{D}^* \boldsymbol{\Omega}^* \mathbf{D}^{*\intercal}),$$
$$p(\mathbf{y}) = \Phi_n(\mathbf{D}\boldsymbol{\xi}; \mathbf{I}_n + \mathbf{D}\boldsymbol{\Omega}\mathbf{D}^\intercal).$$

Recalling our discussion in Section 2.1, $p(\mathbf{y})$ is the marginal likelihood for the observed data and can be expressed as

$$\begin{aligned} p(\mathbf{y}) &= \int \Phi_n(\mathbf{D}\mathbf{f}(\mathbf{X}); \mathbf{I}_n) \phi_n(\mathbf{f}(\mathbf{X}) - \boldsymbol{\xi}; \boldsymbol{\Omega}) \mathrm{d}\mathbf{f}(\mathbf{X}) \\ &= \mathbb{E}[\Phi_n(\mathbf{D}(\mathbf{f}(\mathbf{X}) - \boldsymbol{\xi}) + \mathbf{D}\boldsymbol{\xi}; \mathbf{I}_n)], \end{aligned}$$

where $(\mathbf{f}(\mathbf{X}) - \boldsymbol{\xi}) \sim \mathrm{N}_n(\mathbf{0}, \boldsymbol{\Omega})$. Hence, by applying Lemma 1 to this expectation, we obtain

$$\mathbb{E}[\Phi_n(\mathbf{D}(\mathbf{f}(\mathbf{X}) - \boldsymbol{\xi}) + \mathbf{D}\boldsymbol{\xi}; \mathbf{I}_n)] = \Phi_n(\mathbf{D}\boldsymbol{\xi}; \mathbf{I}_n + \mathbf{D}\boldsymbol{\Omega}\mathbf{D}^\intercal).$$

The above result also clarifies Equation (3). The proof of equation $p(y_{n+1} = 1, \mathbf{y}) = \Phi_{n+1}(\mathbf{D}^* \boldsymbol{\xi}^*; \mathbf{I}_{n+1} + \mathbf{D}^* \boldsymbol{\Omega}^* \mathbf{D}^{*\intercal})$ proceeds in a similar manner, after noticing that

$$\begin{aligned} &p(y_{n+1} = 1, \mathbf{y}) \\ &= \int \Phi(f(\mathbf{x}_{n+1})) \Phi_n(\mathbf{D}\mathbf{f}(\mathbf{X}); \mathbf{I}_n) \phi_{n+1}(\mathbf{f}^*(\mathbf{X}) - \boldsymbol{\xi}^*; \boldsymbol{\Omega}^*) \mathrm{d}\mathbf{f}^*(\mathbf{X}) \\ &= \int \Phi_{n+1}(\mathbf{D}^* \mathbf{f}^*(\mathbf{X}); \mathbf{I}_{n+1}) \phi_{n+1}(\mathbf{f}^*(\mathbf{X}) - \boldsymbol{\xi}^*; \boldsymbol{\Omega}^*) \mathrm{d}\mathbf{f}^*(\mathbf{X}) \\ &= \mathbb{E}[\Phi_{n+1}(\mathbf{D}^*(\mathbf{f}^*(\mathbf{X}) - \boldsymbol{\xi}^*) + \mathbf{D}^* \boldsymbol{\xi}^*; \mathbf{I}_{n+1})], \end{aligned}$$

where $\mathbf{f}^*(\mathbf{X}) - \boldsymbol{\xi}^* = [(\mathbf{f}(\mathbf{X})^\intercal, f(\mathbf{x}_{n+1}))^\intercal - \boldsymbol{\xi}^*] \sim \mathrm{N}_{n+1}(\mathbf{0}, \boldsymbol{\Omega}^*)$, with $\boldsymbol{\xi}^*$, $\boldsymbol{\Omega}^*$ and $\mathbf{D}^*$ defined as in Proposition 1. □

Proof of Proposition 2. Recalling the results discussed in Section 2.2, the predictive probability $\mathrm{pr}(y_{n+1} = 1 \mid \mathbf{y})$ can be defined as $\mathbb{E}_{f(\mathbf{x}_{n+1}) \mid \mathbf{y}}[\Phi(f(\mathbf{x}_{n+1}))]$, with $p(f(\mathbf{x}_{n+1}) \mid \mathbf{y})$ being the marginal in the joint conditional density $p(f(\mathbf{x}_{n+1}), \mathbf{f}(\mathbf{X}), \mathbf{z} \mid \mathbf{y})$ which factorizes as $p(f(\mathbf{x}_{n+1}) \mid \mathbf{f}(\mathbf{X})) p(\mathbf{f}(\mathbf{X}) \mid \mathbf{z}) p(\mathbf{z} \mid \mathbf{y})$. Hence, by the law of the total expectation, we have that

$$\begin{aligned} &\mathrm{pr}(y_{n+1} = 1 \mid \mathbf{y}) \\ &\quad = \mathbb{E}_{\mathbf{z} \mid \mathbf{y}}[\mathbb{E}_{\mathbf{f}(\mathbf{X}) \mid \mathbf{z}}(\mathbb{E}_{f(\mathbf{x}_{n+1}) \mid \mathbf{f}(\mathbf{X})}[\Phi(f(\mathbf{x}_{n+1}))])]. \end{aligned}$$

Since $(f(\mathbf{x}_{n+1}) \mid \mathbf{f}(\mathbf{X})) \sim \mathrm{N}(\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}} \mathbf{f}(\mathbf{X}), \sigma^2_{x_{n+1}})$ by (8), we can leverage Lemma 1 above to obtain

$$\begin{aligned} &\mathbb{E}_{f(\mathbf{x}_{n+1}) \mid \mathbf{f}(\mathbf{X})}[\Phi(f(\mathbf{x}_{n+1}))] \\ &\quad = \Phi(\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}} \mathbf{f}(\mathbf{X}); 1 + \sigma^2_{x_{n+1}}). \end{aligned}$$

To conclude the proof note that, by (9), we have $(\mathbf{f}(\mathbf{X}) \mid \mathbf{z}) \sim \mathrm{N}_n(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}} \mathbf{z}, \boldsymbol{\Sigma}_{\mathbf{X}})$. Therefore, further application of Lemma 1 yields

$$\begin{aligned} &\mathbb{E}_{\mathbf{f}(\mathbf{X}) \mid \mathbf{z}}[\Phi(\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}} \mathbf{f}(\mathbf{X}); 1 + \sigma^2_{x_{n+1}})] \\ &\quad = \Phi(\mu_{x_{n+1}} + \mathbf{H}_{x_{n+1}} (\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}} \mathbf{z}); \\ &\qquad\qquad 1 + \sigma^2_{x_{n+1}} + \mathbf{H}_{x_{n+1}} \boldsymbol{\Sigma}_{\mathbf{X}} \mathbf{H}^\intercal_{x_{n+1}}) \end{aligned}$$

as in Proposition 2. □

Acknowledgments

We are grateful to the Editor, the Associate Editor, and the two referees for the precious comments, which helped us in improving the preliminary version of the article.

Funding

This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No: OSR-2018-CRG7-3742.

ORCID

Jian Cao http://orcid.org/0000-0003-1609-4921

References

Albert, J. H., and Chib, S. (1993), "Bayesian Analysis of Binary and Polychotomous Response Data," *Journal of the American Statistical Association*, 88, 669–679. [710]

Arellano-Valle, R. B., and Azzalini, A. (2006), "On the Unification of Families of Skew-Normal Distributions," *Scandinavian Journal of Statistics*, 33, 561–574. [710,713,719]

Azzalini, A., and Capitanio, A. (2014), *The Skew-Normal and Related Families*, Cambridge: Cambridge University Press. [711,713,719]

Benavoli, A., Azzimonti, D., and Piga, D. (2020), "Skew Gaussian Processes for Classification," *Machine Learning*, 109, 1877–1902. [719]

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011), "Algorithms for Hyper-Parameter Optimization," *Advances in Neural Information Processing Systems*, 24, 2546–2554. [718]

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, New York: Springer. [714]

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017), "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, 112, 859–877. [713,714]

Botev, Z. (2017), "The Normal Law Under Linear Restrictions: Simulation and Estimation via Minimax Tilting," *Journal of the Royal Statistical Society*, Series B, 79, 125–148. [710,711,712,713,714,715,716]

Brezger, A., and Lang, S. (2006), "Generalized Structured Additive Regression Based on Bayesian P-splines," *Computational Statistics & Data Analysis*, 50, 967–991. [709]

Cao, J., Genton, M. G., Keyes, D. E., and Turkiyyah, G. M. (2019), "Hierarchical-Block Conditioning Approximations for High-Dimensional Multivariate Normal Probabilities," *Statistics and Computing*, 29, 585–598. [710,711]

Cao, J., Genton, M. G., Keyes, D. E, and Turkiyyah, G. M. (2021), "Exploiting Low Rank Covariance Structures for Computing High-Dimensional Normal and Student-t Probabilities," *Statistics and Computing*, 31, 2. [710,711,712,713,716]

Chen, W., Castruccio, S., Genton, M. G., and Crippa, P. (2018), "Current and Future Estimates of Wind Energy Potential Over Saudi Arabia," *Journal of Geophysical Research: Atmospheres*, 123, 6443–6459. [717]

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010), "BART: Bayesian Additive Regression Trees," *The Annals of Applied Statistics*, 4, 266–298. [709]

Chib, S., and Greenberg, E. (1998), "Analysis of Multivariate Probit Models," *Biometrika*, 85, 347–361. [719]

Chopin, N. (2011), "Fast Simulation of Truncated Gaussian Distributions," *Statistics and Computing*, 21, 275–288. [710,711]

Chopin, N., and Ridgway, J. (2017), "Leave Pima Indians Alone: Binary Regression as a Benchmark for Bayesian Computation," *Statistical Science*, 32, 64–87. [709,710]

Choudhuri, N., Ghosal, S., and Roy, A. (2007), "Nonparametric Binary Regression Using a Gaussian Process Prior," *Statistical Methodology*, 4, 227–243. [709,710]

Chu, W., and Ghahramani, Z. (2005), "Gaussian Processes for Ordinal Regression," *Journal of Machine Learning Research*, 6, 1019–1041. [709,710]

Consonni, G., and Marin, J.-M. (2007), "Mean-Field Variational Approximate Bayesian Inference for Latent Variable Models," *Computational Statistics & Data Analysis*, 52, 790–798. [710]

De Oliveira, V. (2005), "Bayesian Inference and Prediction of Gaussian Random Fields Based on Censored Data," *Journal of Computational and Graphical Statistics*, 14, 95–115. [709,710]

Durante, D. (2019), "Conjugate Bayes for Probit Regression via Unified Skew-Normal Distributions," *Biometrika*, 106, 765–779. [710,713,714,719]

Fasano, A., Rebaudo, G., Durante, D., and Petrone, S. (2021), "A Closed-Form Filter for Binary Time Series," *Statistics and Computing*, 31, 47. [719]

Fasano, A., and Durante, D. (in press), "A Class of Conjugate Priors for Multinomial Probit Models which Includes the Multivariate Normal One," *Journal of Machine Learning Research*. [719]

Fasano, A., Durante, D., and Zanella, G. (in press), "Scalable and Accurate Variational Bayes for High-Dimensional Binary Regression Models," *Biometrika*. [710,714,715,716]

Genton, M. G., Keyes, D. E., and Turkiyyah, G. (2018), "Hierarchical Decompositions for the Computation of High-Dimensional Multivariate Normal Probabilities," *Journal of Computational and Graphical Statistics*, 27, 268–277. [711]

Genz, A. (1992), "Numerical Computation of Multivariate Normal Probabilities," *Journal of Computational and Graphical Statistics*, 1, 141–149. [711]

Gelman, A., Jakulin, A., Pittau, M. G., and Su, Y. S. (2008), "A Weakly Informative Default Prior Distribution for Logistic and Other Regression Models," *Annals of Applied Statistics*, 2, 1360–1383. [709]

Giani, P., Tagle, F., Genton, M. G., Castruccio, S., and Crippa, P. (2020), "Closing the Gap between Wind Energy Targets and Implementation for Emerging Countries," *Applied Energy*, 269, 115085. [717]

Girolami, M., and Rogers, S. (2006), "Variational Bayesian Multinomial Probit Regression with Gaussian Process Priors," *Neural Computation*, 18, 1790–1817. [709,710,715]

Hoffman, M. D., and Gelman, A. (2014), "The No-U-turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," *Journal of Machine Learning Research*, 15, 1593–1623. [710,715,716]

Holmes, C. C., and Held, L. (2006), "Bayesian Auxiliary Variable Models for Binary and Multinomial Regression," *Bayesian Analysis*, 1, 145–168. [710,713,714]

Horrace, W. C. (2005), "Some Results on the Multivariate Truncated Normal Distribution," *Journal of Multivariate Analysis*, 94, 209–221. [714]

Johndrow, J. E., Smith, A., Pillai, N., and Dunson, D. B. (2019), "MCMC for Imbalanced Categorical Data," *Journal of the American Statistical Association*, 114, 1394–1403. [710]

Klein, A., Falkner, S., Bartels, S., Hennig, P., and Hutter, F. (2017), "Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets," *Artificial Intelligence and Statistics*, 54, 528–536. [718]

Kullback, S., and Leibler, R. A. (1951), "On Information and Sufficiency," *The Annals of Mathematical Statistics*, 22, 79–86. [714]

Kuss, M., and Rasmussen, C. E. (2005), "Assessing Approximate Inference for Binary Gaussian Process Classification," *Journal of Machine Learning Research*, 6, 1679–1704. [709,710,713,715,718]

Neal, R. (1999), "Regression and Classification Using Gaussian Process Priors," *Bayesian Statistics*, 6, 475–501. [709,710]

Nelder, J. A., and Wedderburn, R. W. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society*, Series A, 135, 370–384. [709]

Nickisch, H., and Rasmussen, C. E. (2008), "Approximations for Binary Gaussian Process Classification," *Journal of Machine Learning Research*, 9, 2035–2078. [710,713,715,718]

Opper, M., and Winther, O. (2000), "Gaussian Processes for Classification: Mean–Field Algorithms," *Neural Computation*, 12, 2655–2684. [709]

Pakman, A., and Paninski, L. (2014), "Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians," *Journal of Computational and Graphical Statistics*, 23, 518–542. [710,714]

Rasmussen, C. E., and Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, Cambridge, MA: MIT Press. [709,710,713,718]

Riihimäki, J., Jylänki, P., and Vehtari, A. (2013), "Nested Expectation Propagation for Gaussian Process Classification with a Multinomial Probit Likelihood," *Journal of Machine Learning Research*, 14, 75–109. [709,710,713,715]

Shaahid, S., Al-Hadhrami, L. M., and Rahman, M. (2014), "Potential of Establishment of Wind Farms in Western Province of Saudi Arabia," *Energy Procedia*, 52, 497–505. [717]

Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Duda, M. G., Huang, X.-Y., Wang, W., and Powers, J. G. (2008), "A Description of the Advanced Research WRF version 3," *NCAR Techincal Note*, 113, 1–125. [717]

Snoek, J., Larochelle, H., and Adams, R. P. (2012), "Practical Bayesian Optimization of Machine Learning Algorithms," *Advances in Neural Information Processing Systems*, 25, 2951–2959. [718]

Tagle, F., Castruccio, S., Crippa, P., and Genton, M. G. (2019), "A Non-Gaussian Spatio-Temporal Model for Daily Wind Speeds Based on a Multivariate Skew-t Distribution," *Journal of Time Series Analysis*, 40, 312–326. [717]

Trinh, G., and Genz, A. (2015), "Bivariate Conditioning Approximations for Multivariate Normal Probabilities," *Statistics and Computing*, 25, 989–996. [711]

Yip, C. M. A. (2018), "Statistical Characteristics and Mapping of Near-Surface and Elevated Wind Resources in the Middle East," Ph.D. thesis. KAUST. [717]