

Efficient expectation propagation for posterior approximation in high-dimensional probit models

Augusto Fasano^a, Niccolò Anceschi^b, Beatrice Franzolini^c, and Giovanni Rebaudo^{a,d}

^aCollegio Carlo Alberto, Turin, IT; augusto.fasano@carloalberto.org

^bDuke University, Durham, USA; niccolo.anceschi@duke.edu

^cAgency for Science, Technology and Research (A*STAR), Singapore, SG;
beatricef@sics.a-star.edu.sg

^dUniversity of Turin, Turin, IT; giovanni.rebaudo@unito.it

Abstract

Bayesian binary regression is a prosperous area of research due to the computational challenges encountered by currently available methods either for high-dimensional settings or large datasets, or both. In the present work, we focus on the expectation propagation (EP) approximation of the posterior distribution in Bayesian probit regression under a multivariate Gaussian prior distribution. Adapting more general derivations in Anceschi et al. (2023), we show how to leverage results on the extended multivariate skew-normal distribution to derive an efficient implementation of the EP routine having a per-iteration cost that scales linearly in the number of covariates. This makes EP computationally feasible also in challenging high-dimensional settings, as shown in a detailed simulation study.

Keywords: Probit Model, Expectation Propagation, Bayesian Inference, Extended Multivariate Skew-Normal Distribution

1. Introduction and literature review

The past few years have seen florid research in Bayesian inference for the probit model [4; 8] as well as its extensions to dynamic [7; 6] and multinomial [5; 9; 10] settings and beyond [1; 11]. This has been driven, among others, by computational challenges that may arise in high-dimensional settings. See [3] for an excellent review of Bayesian computations for binary regression. Here, we focus on the expectation propagation (EP) approximation of the posterior of the Bayesian probit model

$$y_i \mid \beta \stackrel{\text{ind}}{\sim} \text{BERN}(\Phi(x_i^\top \beta)), \quad i = 1, \dots, n, \quad (1)$$

$$\beta \sim N_p(\mathbf{0}, \nu^2 \mathbf{I}_p),$$

where $\beta \in \mathbb{R}^p$ is the unknown vector of parameters, $x_i \in \mathbb{R}^p$ is the covariate vector associated with observation i and \mathbf{I}_p denotes the identity matrix of dimension p . $\Phi(t)$ denotes instead the cumulative distribution function of a standard Gaussian random variable evaluated at t . Similarly, $\phi_p(\mathbf{t}, \mathbf{S})$ will denote the density of a p -variate Gaussian random variable with mean $\mathbf{0}$ and covariance matrix \mathbf{S} , evaluated at \mathbf{t} . [4] showed that the posterior distribution for model (1) is a unified skew-normal (SUN) and that, thanks to characterization properties of the SUN family, one can obtain i.i.d. samples from it via a linear combination of p -variate Gaussian samples and n -variate truncated Gaussian samples. As the computational bottleneck is represented by the truncated normal component, such i.i.d. sampler is well-suited for high-dimensional problems with small-to-moderate sample sizes but may become computationally hard for larger sample sizes. To overcome such limitation, [8] developed a partially-factorized variational (PFM-VB) approximation of the posterior distribution which, for any fixed n , converges to the true posterior distribution as p diverges. Crucially, PFM-VB does not require dealing with any multivariate truncated Gaussian since the corresponding density component is replaced with a product of univariate truncated Gaussian densities, which do not represent a computational problem. This approximation has a pre-processing cost of $\mathcal{O}(pn \cdot \min\{p, n\})$ and cost-per-iteration of $\mathcal{O}(n \cdot \min\{p, n\})$, making it computationally tractable also in large p and large n settings. Empirically, the approximate posterior moments closely match the ones obtained via i.i.d. sampling for $p \geq 2n$. The possible over-shrinkage of the posterior moments towards zero for smaller p motivates the investigation of efficient implementations of other approximation techniques that may be more accurate in those settings, like EP, at the price of a higher computational cost. Adapting more general results obtained for a broad class of models in [1], we show how the EP routine for posterior inference under the multivariate Gaussian prior in (1) can be implemented at per-iteration-cost of $\mathcal{O}(pn \cdot \min\{p, n\})$, which, although higher than the one of PFM-VB, improves over the cost $\mathcal{O}(p^2n)$ reported in [3], leading to sensible computational advantages and making EP computationally feasible also in settings with p of the order of tens of thousands. Considering the goodness of the EP approximation [1; 3], the possibility to extend the number of scenarios where it can be effectively implemented represents a major contribution to Bayesian binary regression computations.

2. Expectation propagation for the probit model

In this section, we present an implementation of EP for the probit model (1) which leverages results on multivariate extended skew-normal (SN) random variables (see [2]). Calling $\mathbf{y} = (y_1, \dots, y_n)$, in EP we approximate $p(\beta \mid \mathbf{y})$ with $q(\beta) \propto \prod_{i=0}^n q_i(\beta)$, where $q_0(\beta), \dots, q_n(\beta)$ are probability density functions and, in particular, $q_0(\beta) = p(\beta)$ and $q_i(\beta) \propto \exp\{-\frac{1}{2}\beta^\top \mathbf{Q}_i \beta + \beta^\top \mathbf{r}_i\}$ for $i = 1, \dots, n$. Hence, writing $q_0(\beta) \propto \exp\{-\frac{1}{2}\beta^\top \mathbf{Q}_0 \beta + \beta^\top \mathbf{r}_0\}$, with $\mathbf{r}_0 = \mathbf{0}$ and $\mathbf{Q}_0 = \nu^{-2} \mathbf{I}_p$, we immediately note that $q(\beta) = \phi_p(\beta - \mathbf{Q}^{-1} \mathbf{r}, \mathbf{Q}^{-1})$, where $\mathbf{r} = \sum_{i=0}^n \mathbf{r}_i$, $\mathbf{Q} = \sum_{i=0}^n \mathbf{Q}_i$.

EP proceeds by updating each site $i = 1, \dots, n$ (we do not update the site of the prior), by iteratively matching the first two moments of the global approximation $q(\beta)$ and the hybrid distribution

$$h_i(\beta) \propto p(y_i \mid \beta) \prod_{j \neq i} q_j(\beta) = \Phi((2y_i - 1)x_i^\top \beta) \prod_{j \neq i} q_j(\beta). \quad (2)$$

To compute the moments of (2), instead of proceeding as [3], we can exploit the fact that some easy algebraic manipulations show that (2) is the kernel of a multivariate extended skew-normal distribution

$\text{SN}_p(\boldsymbol{\xi}_i, \boldsymbol{\Omega}_i, \boldsymbol{\alpha}_i, \tau_i)$ (see [2]), with

$$\begin{aligned}\boldsymbol{\xi}_i &= \mathbf{Q}_{-i}^{-1} \mathbf{r}_{-i}, \quad \boldsymbol{\Omega}_i = \mathbf{Q}_{-i}^{-1}, \\ \boldsymbol{\alpha}_i &= (2y_i - 1)\boldsymbol{\omega}_i \mathbf{x}_i, \quad \tau_i = (2y_i - 1)(1 + \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i)^{-1/2} \mathbf{x}_i^\top \boldsymbol{\xi}_i,\end{aligned}$$

where $\mathbf{Q}_{-i} = \sum_{j \neq i} \mathbf{Q}_j$, $\mathbf{r}_{-i} = \sum_{j \neq i} \mathbf{r}_j$ and $\boldsymbol{\omega}_i = [\text{diag}(\boldsymbol{\Omega}_i)]^{1/2}$.

After noticing this, exploiting formulae (5.71) and (5.72) in [2], we can immediately obtain the first two moments of $h_i(\boldsymbol{\beta})$:

$$\begin{aligned}\boldsymbol{\mu}_{h_i} &= \mathbb{E}_{h_i(\boldsymbol{\beta})}[\boldsymbol{\beta}] = \boldsymbol{\xi}_i + \zeta_1(\tau_i)s_i \boldsymbol{\Omega}_i \mathbf{x}_i \\ \boldsymbol{\Sigma}_{h_i} &= \text{var}_{h_i(\boldsymbol{\beta})}[\boldsymbol{\beta}] = \boldsymbol{\Omega}_i + \zeta_2(\tau_i)s_i^2(\boldsymbol{\Omega}_i \mathbf{x}_i)(\boldsymbol{\Omega}_i \mathbf{x}_i)^\top,\end{aligned}$$

where $s_i = (2y_i - 1)(1 + \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i)^{-1/2}$, $\zeta_1(x) = \phi(x)/\Phi(x)$ and $\zeta_2(x) = -\zeta_1(x)^2 - x\zeta_1(x)$. Hence, when updating site i , the EP moment-matching condition implies that the updated quantities $\mathbf{r}_i^{\text{NEW}}$ and $\mathbf{Q}_i^{\text{NEW}}$ must be such that

$$\begin{cases} (\mathbf{Q}_{-i} + \mathbf{Q}_i^{\text{NEW}})^{-1}(\mathbf{r}_{-i} + \mathbf{r}_i^{\text{NEW}}) = \boldsymbol{\mu}_{h_i} \\ (\mathbf{Q}_{-i} + \mathbf{Q}_i^{\text{NEW}})^{-1} = \boldsymbol{\Sigma}_{h_i}, \end{cases}$$

from which it immediately follows

$$\begin{cases} \mathbf{r}_i^{\text{NEW}} = (\mathbf{Q}_{-i} + \mathbf{Q}_i^{\text{NEW}})\boldsymbol{\mu}_{h_i} - \mathbf{r}_{-i} \\ \mathbf{Q}_i^{\text{NEW}} = \boldsymbol{\Sigma}_{h_i}^{-1} - \mathbf{Q}_{-i}. \end{cases}$$

The direct computation of $\boldsymbol{\Sigma}_{h_i}^{-1}$ can be avoided since, by Woodbury's identity

$$\begin{aligned}\mathbf{Q}_i^{\text{NEW}} &= \boldsymbol{\Omega}_i^{-1} - \zeta_2(\tau_i)s_i^2(1 + \zeta_2(\tau_i)s_i^2 \mathbf{x}_i^\top \boldsymbol{\Omega}_i \boldsymbol{\Omega}_i^{-1} \boldsymbol{\Omega}_i \mathbf{x}_i)^{-1} \boldsymbol{\Omega}_i^{-1} \boldsymbol{\Omega}_i \mathbf{x}_i \mathbf{x}_i^\top \boldsymbol{\Omega}_i \boldsymbol{\Omega}_i^{-1} - \mathbf{Q}_{-i} \\ &= -\zeta_2(\tau_i)s_i^2(1 + \zeta_2(\tau_i)s_i^2 \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i)^{-1} \mathbf{x}_i \mathbf{x}_i^\top = -(\zeta_2(\tau_i)^{-1}s_i^{-2} + \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i)^{-1} \mathbf{x}_i \mathbf{x}_i^\top \\ &= -\frac{\zeta_2(\tau_i)}{1 + \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i + \zeta_2(\tau_i)\mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i} \mathbf{x}_i \mathbf{x}_i^\top = k_i^{\text{NEW}} \mathbf{x}_i \mathbf{x}_i^\top,\end{aligned}$$

with $k_i^{\text{NEW}} = -\zeta_2(\tau_i)/(1 + \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i + \zeta_2(\tau_i)\mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i)$. Moreover,

$$\begin{aligned}\mathbf{r}_i^{\text{NEW}} &= \mathbf{Q}_{-i} \boldsymbol{\mu}_{h_i} + \mathbf{Q}_i^{\text{NEW}} \boldsymbol{\mu}_{h_i} - \mathbf{r}_{-i} = \mathbf{Q}_{-i} \mathbf{Q}_{-i}^{-1} \mathbf{r}_{-i} + \zeta_1(\tau_i)s_i \mathbf{Q}_{-i} \boldsymbol{\Omega}_i \mathbf{x}_i + \mathbf{Q}_i^{\text{NEW}} \boldsymbol{\mu}_{h_i} - \mathbf{r}_{-i} \\ &= \zeta_1(\tau_i)s_i \mathbf{x}_i + \mathbf{Q}_i^{\text{NEW}} \boldsymbol{\mu}_{h_i} = \zeta_1(\tau_i)s_i \mathbf{x}_i + k_i^{\text{NEW}} \mathbf{x}_i \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{r}_{-i} + k_i^{\text{NEW}} \zeta_1(\tau_i)s_i \mathbf{x}_i \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i \\ &= [\zeta_1(\tau_i)s_i + k_i^{\text{NEW}}(\boldsymbol{\Omega}_i \mathbf{x}_i)^\top \mathbf{r}_{-i} + k_i^{\text{NEW}} \zeta_1(\tau_i)s_i \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i] \mathbf{x}_i = m_i^{\text{NEW}} \mathbf{x}_i,\end{aligned}$$

Algorithm 1: Probit EP - $\mathcal{O}(p^2n)$ cost per iteration

Initialization: $\mathbf{Q} = \nu^{-2} \mathbf{I}_p$; $\mathbf{Q}^{-1} = \nu^2 \mathbf{I}_p$; $\mathbf{r} = \mathbf{0}$; $k_i = 0$ and $m_i = 0$ for $i = 1, \dots, n$.

for t from 1 until convergence **do**

for i from 1 to n do <ul style="list-style-type: none"> $\mathbf{Q}_{-i} = \mathbf{Q} - k_i \mathbf{x}_i \mathbf{x}_i^\top$ $\mathbf{r}_{-i} = \mathbf{r} - m_i \mathbf{x}_i$ $\boldsymbol{\Omega}_i = \mathbf{Q}^{-1} + k_i / (1 - k_i \mathbf{x}_i^\top \mathbf{Q}^{-1} \mathbf{x}_i) (\mathbf{Q}^{-1} \mathbf{x}_i) (\mathbf{Q}^{-1} \mathbf{x}_i)^\top$ $s_i = (2y_i - 1)(1 + \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i)^{-1/2}$ $\tau_i = s_i \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{r}_{-i}$ $k_i = -\zeta_2(\tau_i)/(1 + \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i + \zeta_2(\tau_i)\mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i)$ $m_i = \zeta_1(\tau_i)s_i + k_i(\boldsymbol{\Omega}_i \mathbf{x}_i)^\top \mathbf{r}_{-i} + k_i \zeta_1(\tau_i)s_i \mathbf{x}_i^\top \boldsymbol{\Omega}_i \mathbf{x}_i$ $\mathbf{Q} = \mathbf{Q}_{-i} + k_i \mathbf{x}_i \mathbf{x}_i^\top$ $\mathbf{r} = \mathbf{r}_{-i} + m_i \mathbf{x}_i$ $\mathbf{Q}^{-1} = \boldsymbol{\Omega}_i + \zeta_2(\tau_i)s_i^2(\boldsymbol{\Omega}_i \mathbf{x}_i)(\boldsymbol{\Omega}_i \mathbf{x}_i)^\top$
--

Output: $q(\boldsymbol{\beta}) = \phi_p(\boldsymbol{\beta} - \mathbf{Q}^{-1} \mathbf{r}; \mathbf{Q}^{-1})$

where $m_i^{\text{NEW}} = \zeta_1(\tau_i)s_i + k_i^{\text{NEW}}(\Omega_i\mathbf{x}_i)^T\mathbf{r}_{-i} + k_i^{\text{NEW}}\zeta_1(\tau_i)s_i\mathbf{x}_i^T\Omega_i\mathbf{x}_i$. Hence, we can implement EP by storing only the scalar quantities k_i and m_i , $i = 1, \dots, n$. In practice, they are initialized to zero, so that the initial global approximation is the prior distribution. Combining the above results with Woodbury's identity, we obtain

$$\Omega_i = \mathbf{Q}_{-i}^{-1} = (\mathbf{Q} - k_i\mathbf{x}_i\mathbf{x}_i^T)^{-1} = \mathbf{Q}^{-1} + \frac{k_i}{1 - k_i\mathbf{x}_i^T\mathbf{Q}^{-1}\mathbf{x}_i} (\mathbf{Q}^{-1}\mathbf{x}_i) (\mathbf{Q}^{-1}\mathbf{x}_i)^T,$$

which can be computed avoiding explicit matrix inversions, since \mathbf{Q}^{-1} is known from the beginning. Finally, the update of the inverse of the EP precision matrix is immediate as $(\mathbf{Q}^{\text{NEW}})^{-1} = (\mathbf{Q}_{-i} + \mathbf{Q}_i^{\text{NEW}})^{-1} = \Sigma_{h_i}$.

Putting it all together, we obtain the EP implementation in Algorithm 1. Its core part coincides with the EP derivations presented in [3], and implemented in the `EPprobit` function in the R package `EPGLM`. However, we arrived at it by exploiting results on SNS that leverage more general derivations for a broader class of models presented in [1]. We also avoided the computation of the normalizing constants for the unnormalized densities, Z_i , $i = 1, \dots, n$, which can be used for the computation of the approximate marginal likelihood, since the approximated posterior moments can be computed also without them. Algorithm 1 has per-iteration cost $\mathcal{O}(p^2n)$, which, although avoiding explicit $p \times p$ matrix inversions, might be impractical in high-dimensional settings. Adapting more general results presented in [1], we thus derive in Section in full detail an implementation of EP for the Bayesian probit model having per-iteration cost $\mathcal{O}(pn^2)$.

3. Efficient expectation propagation for large p settings

The crucial part to obtain a per-iteration-cost that is linear in p is to note that we can avoid handling $p \times p$ matrices, as, by close inspection of Algorithm 1, the whole EP routine can be written by working out directly the updates of the p -dimensional vectors $\mathbf{w}_i = \Omega_i\mathbf{x}_i = \mathbf{Q}_{-i}^{-1}\mathbf{x}_i$ and $\mathbf{v}_i = \mathbf{Q}^{-1}\mathbf{x}_i$, $i = 1, \dots, n$. As for the former, we have

$$\begin{aligned} \mathbf{w}_i &= \mathbf{Q}_{-i}^{-1}\mathbf{x}_i = (\mathbf{Q} - \mathbf{Q}_i)^{-1}\mathbf{x}_i = \mathbf{Q}^{-1}\mathbf{x}_i + (1 - k_i\mathbf{x}_i^T\mathbf{Q}^{-1}\mathbf{x}_i)^{-1}k_i(\mathbf{Q}^{-1}\mathbf{x}_i)(\mathbf{Q}^{-1}\mathbf{x}_i)^T\mathbf{x}_i \\ &= \mathbf{v}_i + k_i(1 - k_i\mathbf{x}_i^T\mathbf{v}_i)^{-1}\mathbf{v}_i\mathbf{v}_i^T\mathbf{x}_i = [1 + (1 - k_i\mathbf{x}_i^T\mathbf{v}_i)^{-1}(k_i\mathbf{x}_i^T\mathbf{v}_i)]\mathbf{v}_i = d_i\mathbf{v}_i, \end{aligned}$$

where $d_i = (1 - k_i\mathbf{x}_i^T\mathbf{v}_i)^{-1}$. As for the \mathbf{v}_i 's, each time a site i is updated \mathbf{Q} changes and thus all the \mathbf{v}_j 's, $j = 1, \dots, n$, should be modified accordingly as

$$\begin{aligned} \mathbf{v}_j^{\text{NEW}} &= (\mathbf{Q}^{\text{NEW}})^{-1}\mathbf{x}_j = (\mathbf{Q} - \mathbf{Q}_i + \mathbf{Q}_i^{\text{NEW}})^{-1}\mathbf{x}_j = [\mathbf{Q} + (k_i^{\text{NEW}} - k_i)\mathbf{x}_i\mathbf{x}_i^T]^{-1}\mathbf{x}_j \\ &= [\mathbf{Q}^{-1} - (k_i^{\text{NEW}} - k_i)[1 + (k_i^{\text{NEW}} - k_i)\mathbf{x}_i^T\mathbf{Q}^{-1}\mathbf{x}_i]^{-1}\mathbf{Q}^{-1}\mathbf{x}_i\mathbf{x}_i^T\mathbf{Q}^{-1}]\mathbf{x}_j \\ &= \mathbf{Q}^{-1}\mathbf{x}_j - [(k_i^{\text{NEW}} - k_i)^{-1} + \mathbf{x}_i^T\mathbf{v}_i]^{-1}\mathbf{v}_i\mathbf{x}_i^T\mathbf{v}_j = \mathbf{v}_j - c_i(\mathbf{x}_i^T\mathbf{v}_j)\mathbf{v}_i, \end{aligned}$$

where $c_i = (k_i^{\text{NEW}} - k_i)/(1 + (k_i^{\text{NEW}} - k_i)\mathbf{x}_i^T\mathbf{v}_i)$. Instead of cycling over j , these updates can be performed in block by defining a $p \times n$ matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$. Accordingly, $\mathbf{V}^{\text{NEW}} = \mathbf{V} - c_i\mathbf{v}_i\mathbf{x}_i^T\mathbf{V}$. This operation is the most expensive per site update, being of order $\mathcal{O}(pn)$. Accordingly, each EP iteration has cost $\mathcal{O}(pn^2)$. Contrarily to Algorithm 1, once the procedure has reached convergence we still need to calculate the inverse of the global precision matrix \mathbf{Q}^{-1} . The explicit calculation can be avoided as follows, obtaining a post-processing cost of $\mathcal{O}(p^2n)$. First, $\mathbf{Q} = \mathbf{Q}_0 + \sum_{i=1}^n k_i\mathbf{x}_i\mathbf{x}_i^T = \nu^{-2}\mathbf{I}_p + \mathbf{X}^T\mathbf{K}\mathbf{X}$ with $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$ $\mathbf{K} = \text{diag}(k_1, \dots, k_n)$. Calling $\Lambda = (\mathbf{I}_n + \nu^2\mathbf{K}\mathbf{X}\mathbf{X}^T)^{-1}$, so that $\mathbf{Q}^{-1} = \nu^2\mathbf{I}_p - \nu^4\mathbf{X}^T\Lambda\mathbf{K}\mathbf{X}$, one obtains that $\mathbf{V} = \mathbf{Q}^{-1}\mathbf{X}^T = \nu^2\mathbf{X}^T[\mathbf{I}_n - \nu^2\Lambda\mathbf{K}\mathbf{X}\mathbf{X}^T] = \nu^2\mathbf{X}^T\Lambda[\Lambda^{-1} - \nu^2\mathbf{K}\mathbf{X}\mathbf{X}^T] = \nu^2\mathbf{X}^T\Lambda$ and thus $\mathbf{Q}^{-1} = \nu^2\mathbf{I}_p - \nu^2\mathbf{V}\mathbf{K}\mathbf{X}$. Notice that, if the interest is only in approximate posterior means and variances, this expression for \mathbf{Q}^{-1} allows doing it at reduced post-processing cost of $\mathcal{O}(pn)$. The whole routine is summarized in Algorithm 2.

Algorithm 2: Efficient probit EP for large p - $\mathcal{O}(p \cdot n^2)$ cost per iteration

Initialization: $\mathbf{r} = \mathbf{0}$; $k_i = 0$ and $m_i = 0$ for $i = 1, \dots, n$; $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n] = \nu^2 \mathbf{X}^\top$.

for t from 1 until convergence **do**

for i from 1 to n do	$\mathbf{w}_i = (1 - k_i \mathbf{x}_i^\top \mathbf{v}_i)^{-1} \mathbf{v}_i$ $\mathbf{r}_{-i} = \mathbf{r} - m_i \mathbf{x}_i$ $s_i = (2y_i - 1)(1 + \mathbf{x}_i^\top \mathbf{w}_i)^{-1/2}$ $\tau_i = s_i \mathbf{w}_i^\top \mathbf{r}_{-i}$ $k_i^{\text{NEW}} = -\zeta_2(\tau_i) / (1 + \mathbf{x}_i^\top \Omega_i \mathbf{x}_i + \zeta_2(\tau_i) \mathbf{x}_i^\top \Omega_i \mathbf{x}_i)$ $m_i = \zeta_1(\tau_i) s_i + k_i^{\text{NEW}} (\Omega_i \mathbf{x}_i)^\top \mathbf{r}_{-i} + k_i^{\text{NEW}} \zeta_1(\tau_i) s_i \mathbf{x}_i^\top \Omega_i \mathbf{x}_i$ $k_i = k_i^{\text{new}}$ $\mathbf{r} = \mathbf{r}_{-i} + m_i \mathbf{x}_i$ $\mathbf{V} = \mathbf{V} - \mathbf{v}_i [(k_i^{\text{NEW}} - k_i) / (1 + (k_i^{\text{NEW}} - k_i) \mathbf{x}_i^\top \mathbf{v}_i)] \mathbf{x}_i^\top \mathbf{V}$
--	---

$\mathbf{Q}^{-1} = \nu^2 \mathbf{I}_p - \nu^2 \mathbf{V} \mathbf{K} \mathbf{X}$

Output: $q(\beta) = \phi_p(\beta - \mathbf{Q}^{-1} \mathbf{r}; \mathbf{Q}^{-1})$

4. Simulation study

We conclude with a simulation study where probit regression is applied to multiple simulated datasets, with $n = 100$ and $p = 50, 100, 200, 400$ and 800 . We investigate the performances of EP when the efficient implementations presented in Algorithm 1 and Algorithm 2 are used when $p < n$ and $p \geq n$, respectively. Such implementation, denoted EP-EFF in the following, is compared with PFM-VB in terms of running time and quality of the approximation. The latter is measured by the median absolute difference between the approximate posterior means and standard deviations and the ones computed via 2000 i.i.d. samples, for $\nu^2 = 25$. The moderate sample size is taken so that the i.i.d. sampler is computationally efficient, but the approximate methods could be used in more challenging settings, as in all scenarios they both give almost immediate outputs. To show the computational gains with respect to standard EP implementations, we also compare the running time needed to obtain the EP approximation with the R function `EPprobit` from the package `EPGLM`, which implements the EP derivations reported in [3]. As it emerges from Table 1, EP-EFF leads to a dramatic reduction of the computational effort with respect to the standard `EPprobit` in high dimensions. This results in a drop of the running time by more than three orders of magnitude in the setting $p = 800$, with a computational gain increasing with p , as expected. The EP-EFF running times, although generally much lower than the ones of `EPprobit`, are still higher than the ones of PFM-VB in most cases. Nevertheless, if one looks at the quality of the approximation of the two posterior moments in Figure 1, EP-EFF gives consistently accurate approximations across different dimensions of p , while PFM-VB gets similar accuracy for $p \gtrsim 2n$. This shows the importance of developing efficient implementations for EP like the ones in this paper, so make it computationally feasible in challenging high-dimensional settings where routine implementations are impractical. Code can be found at <https://github.com/augustofasano/EPprobit-SN>.

Table 1: Running time, in seconds, to compute posterior means and standard deviations with the EP approximation as in Algorithms 1 and 2 (EP-EFF), with the EP approximation computed via the R function `EPprobit` (`EPprobit`) and with the PFM-VB approximation (PFM-VB) for probit regression with $n = 100$ and $\nu^2 = 25$.

		p				
		50	100	200	400	800
Running time (seconds)	Method	0.11	0.02	0.03	0.05	0.09
	EPprobit	0.07	0.42	3.18	24.36	140.24
	PFM-VB	0.11	0.06	0.01	0.01	0.01

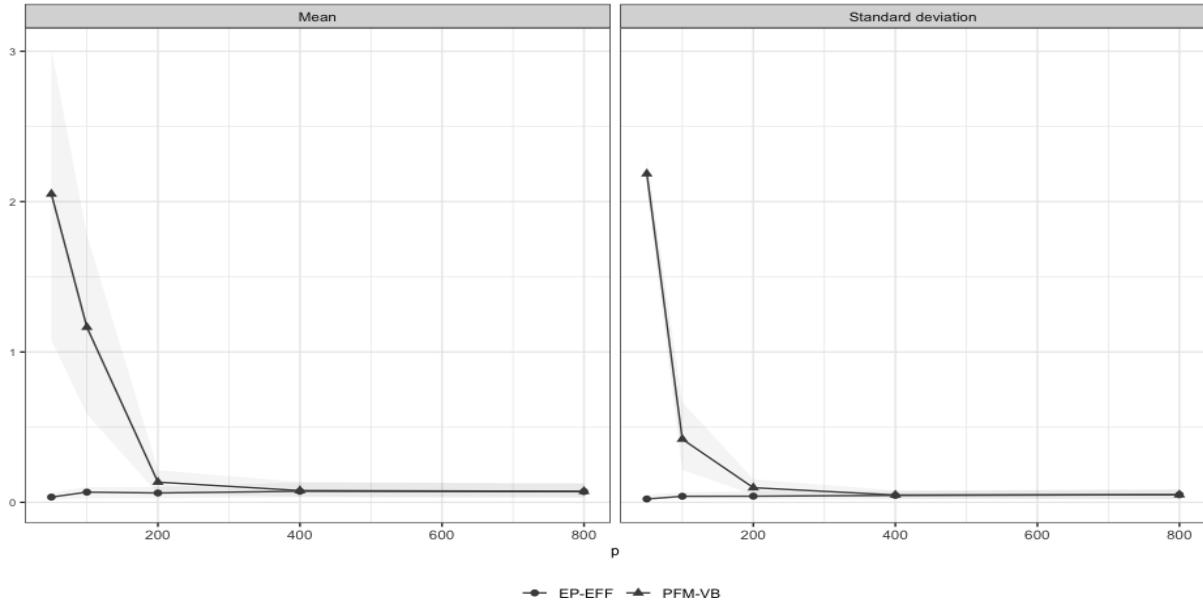


Figure 1: For varying p , median absolute difference between the p posterior means and standard deviations resulting from 2000 i.i.d. samples and the ones arising from EP-EFF and PFM-VB for probit regression with $n = 100$ and $\nu^2 = 25$. Grey areas denote the first and third quartiles.

Acknowledgments The authors wish to thank D. Durante for carefully reading a preliminary version of this manuscript and providing insightful comments.

References

- [1] Anceschi, N., Fasano, A., Durante, D. and Zanella, G.: Bayesian conjugacy in probit, tobit, multinomial probit and extensions: a review and new results. *Journal of the American Statistical Association* [in press] (2023)
- [2] Azzalini, A. and Capitanio, A.: *The Skew-Normal and Related Families*. Cambridge University Press (2014)
- [3] Chopin, N. and Ridgway, J.: Leave Pima Indians alone: binary regression as a benchmark for Bayesian computation. *Statistical Science*, **32**, 64–87 (2017)
- [4] Durante, D.: Conjugate Bayes for probit regression via unified skew-normal distributions. *Biometrika*, **106**, 765–779 (2019)
- [5] Fasano, A. and Durante, D.: A class of conjugate priors for multinomial probit models which includes the multivariate normal one. *Journal of Machine Learning Research*, **23**, 1–16 (2022)
- [6] Fasano, A. and Rebaudo, G.: Variational inference for the smoothing distribution in dynamic probit models. *Book of Short Papers - SIS 2021*, 1076-1081 (2021)
- [7] Fasano, A., Rebaudo, G., Durante, D., and Petrone, S.: A closed-form filter for binary time series. *Statistics and Computing*, **31**, 1–20 (2021)
- [8] Fasano, A., Durante, D. and Zanella, G.: Scalable and accurate variational Bayes for high-dimensional binary regression models. *Biometrika*, **109**, 901–919 (2022)
- [9] Fasano, A., Rebaudo, G. and Anceschi, N.: Bayesian inference for the multinomial probit model under Gaussian prior distribution. *Book of Short Papers - SIS 2022*, 871–876 (2022)
- [10] Loaiza-Maya, R. and Nibbering, D.: Fast variational Bayes methods for multinomial probit models. *Journal of Business & Economic Statistics* [online version] (2022)
- [11] Loaiza-Maya, R., Smith, M. S., Nott, D. J. and Danaher, P. J.: Fast and accurate variational inference for models with many latent variables. *Journal of Econometrics*, **230**, 229–362 (2022)