A variational inference framework for inverse problems

Luca Maestrini [a],[*], Robert G. Aykroyd [b], Matt P. Wand [c]

[a] Research School of Finance, Actuarial Studies and Statistics, Australian National University, Building 26C Kingsley Street, Canberra, Acton 2601, Australian Capital Territory, Australia

[b] Department of Statistics, School of Mathematics, University of Leeds, Leeds, LS2 9JT, United Kingdom

[c] School of Mathematical and Physical Sciences, University of Technology Sydney, P.O. Box 123 Broadway, Sydney, Ultimo 2007, New South Wales, Australia

ARTICLE INFO

ABSTRACT

A framework is presented for fitting inverse problem models via variational Bayes approximations. This methodology guarantees flexibility to statistical model specification for a broad range of applications, good accuracy and reduced model fitting times. The message passing and factor graph fragment approach to variational Bayes that is also described facilitates streamlined implementation of approximate inference algorithms and allows for supple inclusion of numerous response distributions and penalizations into the inverse problem model. Models for one- and two-dimensional response variables are examined and an infrastructure is laid down where efficient algorithm updates based on nullifying weak interactions between variables can also be derived for inverse problems in higher dimensions. An image processing application and a simulation exercise motivated by biomedical problems reveal the computational advantage offered by efficient implementation of variational Bayes over Markov chain Monte Carlo.

1. Introduction

Inverse problems are essentially statistical regression problems where a response depending on a number of parameters is measured and the goal is to interpret the parameter estimates, rather then predict the outcome. Stable fitting of inverse problems is crucial but this is generally hindered by a large number of parameters and the presence of predictors which are highly correlated.

Let y denote a vector of data and suppose this data is related to a vector of unknown parameters x to estimate by a linear regression problem $E(y) = Kx$, where K is given, or a nonlinear one such that $E(y) = g(x)$, where g is a known function. From a Bayesian perspective, model fitting can be performed by placing a prior on x to then find the maximum a posteriori estimate $\hat{x} = \mathrm{argmax}_x\, p(x|y)$. This appears straightforward in principle, however, in typical applications, inverse problems may be ill-posed in a sense that either the solution does not exist, is not unique or does not depend smoothly on the data, as small noise variations can produce significantly different estimates (Hadamard, 1902). A remedy is to introduce a penalization in the model formulation and use Bayesian hierarchical models, but these can be slow to fit via standard Markov chain Monte Carlo methods. To overcome this issue, we propose and study variational Bayes methods. The direct and message passing approaches to variational Bayes we examine facilitate inverse problem fitting in Bayesian settings with reduced computational times.

The use of variational Bayesian methods for inverse problems has been shown in the literature concerning neural source reconstruction, including Sato et al. (2004), Kiebel et al. (2008), Wipf and Nagarajan (2009) and Nathoo et al. (2014). Approximate

inference methods motivated by a broader class of inverse problem applications are in their infancy. A small, growing, literature includes McGrory et al. (2009), Gehre and Jin (2014) and Guha et al. (2015). Arridge et al. (2018) and Zhang et al. (2019) respectively study usage of Gaussian variational approximations and expectation propagation to fit inverse problems models with Poisson responses. Tonolini et al. (2020) propose a framework to train variational inference for imaging inverse problems exploiting existing image data. Agrawal et al. (2022) study variational inference for inverse problems with gamma hyperpriors. Povala et al. (2022) present a stochastic variational Bayes approach based on sparse precision matrices.

The state-of-the-art in approximate inference for inverse problems is to derive and code algorithm updates from scratch each time a model is modified. The message passing on factor graph fragment approach to variational Bayes we suggest in this work overcomes the issue. Wand (2017) has spearheaded adoption of this approach to fast approximation inference in regression-type models via variational message passing (VMP). In the same spirit, we lay down similar infrastructure for inverse problems and propose VMP as an alternative to the more common mean field variational Bayes (MFVB). We show how to perform approximate inference by combining algorithms for single factor graph components, or *fragments*, that arise from inverse problem models. The resultant factor graph fragments facilitate streamlined implementation of fast approximate algorithms and form the basis for software development for use in applications. The factor graph fragment paradigm allows for easy incorporation of different penalization structures in the model or changes to the distribution of the outcome variable. In fact, VMP on factor graph fragments is such that the corresponding algorithms only need to be derived once for a particular fragment and can be used for any arbitrarily complex model including such a fragment. Hence dramatically reducing set-up overheads as well as providing fast implementation.

Motivated by a real biomedical problem, we identify a base inverse problem model and describe how to efficiently perform MFVB and VMP. The application we show concerns medical positron emission tomography imaging where the raw data is processed for image enhancement. The data were collected to illustrate a small animal imaging system which can be used in biotechnology and pre-clinical medical research to help detect tumors or organ dysfunctions. An application to two-dimensional deconvolution problems motivated by an archaeological exploration is also embarked upon the base framework by varying the response and penalization distributional assumptions. This is illustrated in the supplementary material.

1.1. Overview of the article

Section 2 defines a reference inverse problem model for illustrating the methodology in use and our computational developments. The variational approximation engine for inverse problem fitting and inference is introduced in Section 3. Section 4 examines strategies to streamline variational inference algorithms. An application to real biomedical data is treated in Section 5. The same section reports results from a study involving simulations which resemble the analyzed biomedical dataset. The supplementary material provides an illustration concerning archaeological data performed via VMP, where the Normal response and Laplace penalization of the base model are replaced by a Skew Normal distribution for the outcome variable and a Horseshoe penalization. Concluding remarks and extensions are discussed in Section 6.

Before setting up our reference linear inverse problem model and presenting variational algorithms for approximate model fitting we introduce some useful notation.

1.2. Useful notation

For a matrix A of size $d_1 \times d_2$, $\text{vec}(A)$ is the $d_1 d_2 \times 1$ vector obtained by stacking the columns of A underneath each other in order from left to right. If a is a $(d_1 d_2) \times 1$ vector then $\text{vec}_{d_1 \times d_2}^{-1}(a)$ is the $d_1 \times d_2$ matrix such that $\text{vec}\left\{\text{vec}_{d_1 \times d_2}^{-1}(a)\right\} = a$; when the vec operator inverse produces a square matrix the subscript is omitted. Vectors of d zeros or ones are respectively denoted by $\mathbf{0}_d$ and $\mathbf{1}_d$.

2. Base inverse problem model

We consider linear inverse problems having the following formulation:

$$y = Kx + \varepsilon, \quad \varepsilon \sim N(\mathbf{0}, \sigma^2 I), \tag{1}$$

where y is an $m \times 1$ vector of observed data, K is a matrix acting as a linear operator of size $m \times m$, x is a $m \times 1$ vector of unknown parameters and ε is a Normal error vector of length m. For ease of illustration, we focus on the case where the vectors y and x have equal length m and therefore K is a square matrix. Nevertheless, the methodology presented here can be adapted to the situation in which y has length n different from and typically smaller than the length m of x. Motivated by our biomedical application, we focus on a particular type of forward problem where K is a kernel matrix. Another formulation of K is discussed in the application to archaeological data treated in the supplementary material.

We assume that the vector of observations y has a one-to-one correspondence with the vector of parameters $x = (x_1, \ldots, x_m)$ to be estimated. For simplicity, here we only model first nearest neighbour interactions, or differences, between elements of x. If these elements are identified by a system of coordinates in two dimensions, then first nearest neighbour interactions are the differences between one parameter and those in adjacent locations on the horizontal and vertical coordinates of the parameter.

Suppose the aim is to study a linear inverse problem in a Bayesian framework according to the model

$$
\begin{aligned}
y_i | \boldsymbol{x}, \sigma_\varepsilon^2 &\overset{\text{ind.}}{\sim} N\left((\boldsymbol{K}\boldsymbol{x})_i, \sigma_\varepsilon^2\right), \quad i = 1, \ldots, m, \\
(\boldsymbol{x}_\Delta)_j | b_j, \sigma_x^2 &\overset{\text{ind.}}{\sim} N\left(0, \sigma_x^2 / b_j\right), \quad b_j \overset{\text{ind.}}{\sim} \text{Inverse-}\chi^2(2, 1), \quad j = 1, \ldots, d, \\
\sigma_\varepsilon^2 | a_\varepsilon &\sim \text{Inverse-}\chi^2\left(1, 1/a_\varepsilon\right), \quad a_\varepsilon \sim \text{Inverse-}\chi^2\left(1, 1/A_\varepsilon^2\right), \\
\sigma_x^2 | a_x &\sim \text{Inverse-}\chi^2\left(1, 1/a_x\right), \quad a_x \sim \text{Inverse-}\chi^2\left(1, 1/A_x^2\right),
\end{aligned}
\tag{2}
$$

where $\boldsymbol{K}$ is a matrix of size $m \times m$, $\boldsymbol{x}_\Delta$ is a vector of d differences between pairs of elements in $\boldsymbol{x}$ and $A_\varepsilon, A_x > 0$ are user-specified hyperparameters. The auxiliary variables a_ε and a_x generate Half-Cauchy$\left(A_\varepsilon\right)$ and Half-Cauchy$\left(A_x\right)$ priors on the scale parameters σ_ε and σ_x, respectively. Specifically, the density function of a random variable $\sigma > 0$ having a Half-Cauchy(A) distribution is $p(\sigma) = 2/[A\pi\{1 + (\sigma/A)^2\}]$, with $A > 0$. For problems that only contemplate first nearest neighbour differences, the scalar d coincides with the number of unique up to sign differences between pairs of elements of $\boldsymbol{x}$ coming from adjacent locations. In the one-dimensional case, $\boldsymbol{x}$ can be interpreted as a vector matching m spatial locations on a line and the number of differences between adjacent locations will be $d = m - 1$. Model (2) also encompasses higher-dimensional problems. In bidimensional settings, $\boldsymbol{x}$ can be conveniently expressed as the vectorization of a grid, or matrix, of pixels $\boldsymbol{X}$ by setting $\boldsymbol{x} = \text{vec}(\boldsymbol{X})$. If $\boldsymbol{X}$ has size $m_1 \times m_2$, the first nearest neighbour differences are $d = m_1(m_2 - 1) + m_2(m_1 - 1)$. In the simple example of Fig. 1 where $\boldsymbol{X}$ is of size 3×4, the number of horizontal and vertical differences are respectively 9 and 8, giving $d = 17$ differences in total. In a similar vein, the model can be applied to three-dimensional problems by letting $\boldsymbol{x}$ be the vectorization of voxel-type data.

The distributional assumption on $\boldsymbol{x}_\Delta$ in model (2) can be conveniently re-expressed as

$$
\boldsymbol{L}\boldsymbol{x} | \boldsymbol{b}, \sigma_x^2 \sim N\left(\boldsymbol{0}_d, \sigma_x^2 \text{diag}(\boldsymbol{b})^{-1}\right),
\tag{3}
$$

where $\boldsymbol{L}$ is some $d \times m$ *contrast matrix* such that $\boldsymbol{x}_\Delta \equiv \boldsymbol{L}\boldsymbol{x}$ and $\boldsymbol{b} = (b_1, \ldots, b_d)$. For instance in one-dimensional problems contemplating first nearest neighbour interactions, the contrast matrix can be defined as

$$
\boldsymbol{L}_{1D} \equiv \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix},
\tag{4}
$$

i.e. as the $(m-1) \times m$ matrix such that

$$
\boldsymbol{x}_\Delta = \boldsymbol{L}_{1D}\boldsymbol{x} = \left[x_2 - x_1, x_3 - x_2, \ldots, x_m - x_{m-1}\right]^T,
$$

where $m - 1$ is the number of unique up to sign differences between adjacent elements of $\boldsymbol{x}$. In practice there is no need to compute a contrast matrix, although for deriving variational algorithms it is useful to carry $\boldsymbol{L}$ around. As for the matrix defined in (4), it is convenient to design $\boldsymbol{L}$ as a matrix whose number of rows and columns are respectively equal to the number of differences d and the length of $\boldsymbol{x}$, m, also in higher-dimensional problems. Model (2) also incorporates a Laplace penalization that originates from the auxiliary variables $b_j > 0$, $j = 1, \ldots, d$, and the following result.

Result 1. *Let x and b be random variables such that*

$$
x \,|\, b \sim N\left(0, \sigma^2/b\right) \quad \text{and} \quad b \sim \text{Inverse-}\chi^2(2, 1), \quad \text{with} \quad \sigma > 0.
$$

Then $x \sim \text{Laplace}(0, \sigma)$.

The choice of imposing priors on the difference $\boldsymbol{x}_\Delta$ is mainly motivated by the two-dimensional applications on biomedical and archaeological imaging considered in this work. Such a choice gives adequate smoothing for flat regions (i.e., it increases image deblurring), but it may oversmooth discontinuities (see Section 3.2 of Aykroyd et al., 2001, for a detailed discussion on this and remedies). We focus on first-neighbour differences to illustrate efficient computation of the variational algorithm updates through removal of the contrast matrix $\boldsymbol{L}$ as per Section 4. Another widely used neighbourhood system is the second order one, which is based on the eight nearest neighbours (e.g., Green, 1990) and also provides ground for efficient algorithm implementations.

If the dimension of $\boldsymbol{x}$ increases, the number of first neighbour differences increases and reduces computational efficiency. This issue is often solved in practice by partitioning the surface where the observations are collected into smaller regions. Sometimes this is also done to facilitate the application of the inverse problem to irregular (e.g., non-rectangular) surfaces. For example, when large archaeological fields are explored it is standard practice to divide the area into grids and examine each grid as soon as it has been surveyed, and the inverse problem reconstruction of each grid is typically extended to part of the neighbouring grids in each direction to get a smoother reconstruction (Aykroyd et al., 2001, Section 5).

The design of the matrix $\boldsymbol{K}$ varies according to the inverse problem characteristics. As illustration and for later use on real biomedical data we consider a Gaussian kernel matrix $\boldsymbol{K}$ and show its application to a simple unidimensional problem. If $\boldsymbol{y}$ is a vector of m recordings from a unidimensional space having one-to-one correspondence with $\boldsymbol{x}$, the (i, j)th entry of a Gaussian kernel matrix $\boldsymbol{K}$ is given as follows:
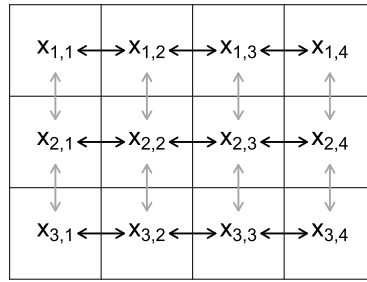
Fig. 1. Example of a 3×4 grid of pixels with 9 horizontal differences marked by black arrows and 8 vertical differences in grey. The total number of first order nearest neighbour differences is 17.
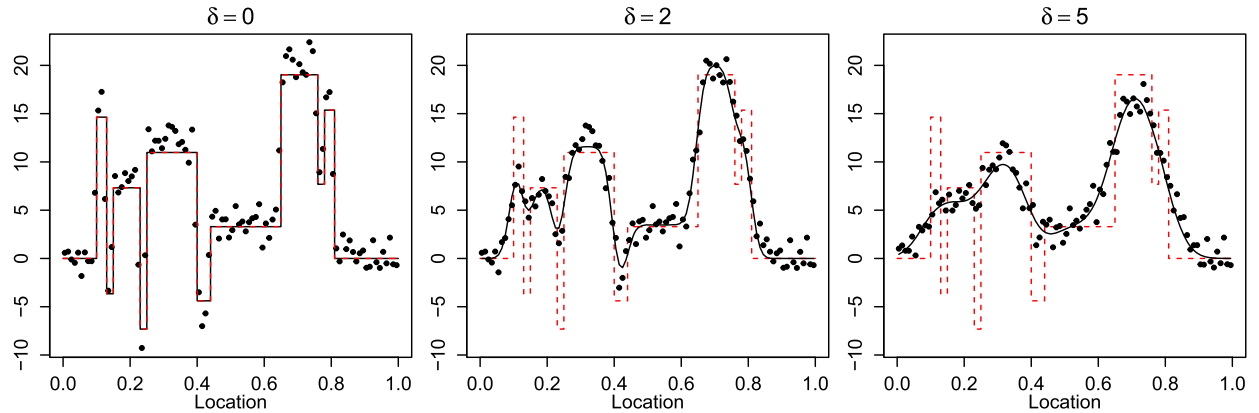

Fig. 2. Data points obtained through the Blocks test function (red dashed line) along with blurred test function (solid line) for different levels of blur: no blur ($\delta = 0$), moderate blur ($\delta = 2$) and large blur ($\delta = 5$). (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

$$K_{ij} = (2\pi\delta^2)^{-1/2} \exp\{-(i-j)^2/(2\delta^2)\}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, m, \tag{5}$$

where $\delta > 0$ is a parameter that governs the amount of blur. Here δ is assumed to be fixed, but it can be estimated within the variational framework by imposing a prior and selected using, for example, a posterior mean estimate. Since this parameter is required to be positive, sensible choices of prior are gamma and inverse gamma distributions (Weir, 1997; Aykroyd et al., 2001). Alternatively, to limit computational costs one could first obtain a discrete approximation of the posterior of δ for a discrete grid of δ values.

To illustrate the effect of variations in δ, consider the *Blocks* test function (Donoho and Johnstone, 1994; Nason, 2008) from the `wavethresh` package (Nason, 2022) available in R (R Core Team, 2023). The piecewise constant nature of this function makes estimation a very challenging problem especially when tackled as an inverse problem, but it is well motivated by stratigraphy problems in archaeology (Allum et al., 1999; Aykroyd et al., 2001). Fig. 2 shows three examples produced through (1) and the Gaussian kernel (5), with $m = 100$, $\sigma = 1$ and $\delta = 0, 2, 5$. In each plot the red dashed line shows the true Blocks function to estimate. In the plot corresponding to $\delta = 0$ blurring is not added to the generation process and the data points are randomly scattered around the true function. Blurring is introduced when $\delta > 0$ and points scatter around a rounded solid line. The cases where $\delta = 2$ and $\delta = 5$ respectively correspond to moderate and large blurring of the underlying true function and hence moderate and difficult inverse problems.

In two dimensions, assuming the observed data is stored in a matrix $\boldsymbol{Y}$, each element of a Gaussian kernel matrix $\boldsymbol{K}$ links an element of $\boldsymbol{y} = \text{vec}(\boldsymbol{Y})$ to one of $\boldsymbol{x} = \text{vec}(\boldsymbol{X})$. The entry of $\boldsymbol{K}$ corresponding to a pair $(Y_{ij}, X_{i'j'})$ is given by

$$(2\pi\delta^2)^{-1} \exp[-\{(i-i')^2 + (j-j')^2\}/(2\delta^2)], \quad \text{for } \delta > 0. \tag{6}$$

The effect of δ on blurring is analogous to the one described for unidimensional problems.

3. Model fitting via variational methods

In this section, we study variational Bayes approximations, specifically MFVB and VMP, for fitting the base model (2). Emphasis is placed onto the message passing on factor graph fragment prescription, which provides the infrastructure for compartmentalized and scalable algorithm implementations.

For studying MFVB and VMP note that the joint density function of all the random variables and random vectors in model (2) admits the following factorization:

$$p\left(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}, \sigma_\varepsilon^2, \sigma_x^2, a_\varepsilon, a_x\right) = p\left(\boldsymbol{y}|\boldsymbol{x}, \sigma_\varepsilon^2\right) p\left(\boldsymbol{x}|\boldsymbol{b}, \sigma_x^2\right) p\left(\boldsymbol{b}\right) p\left(\sigma_\varepsilon^2|a_\varepsilon\right) p\left(\sigma_x^2|a_x\right) p\left(a_\varepsilon\right) p\left(a_x\right). \tag{7}$$
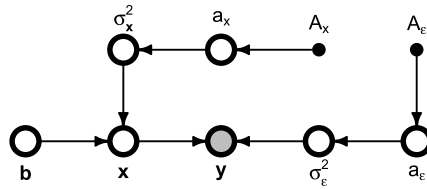
Fig. 3. Directed acyclic graph representation of the Normal response model with Laplace penalization in (2). The shaded circle corresponds to the observed data. The unshaded circles correspond to model parameters and auxiliary variables. The small solid circles correspond to hyperparameters.

Both variational inference procedures are based upon approximating the joint posterior density function through a product of approximating density functions. A possible mean field restriction on the joint posterior density function of all parameters in (2) is

$$p\left(\boldsymbol{x}, \boldsymbol{b}, \sigma_\varepsilon^2, \sigma_x^2, a_\varepsilon, a_x | \boldsymbol{y}\right) \approx q(\boldsymbol{x}) q\left(\sigma_\varepsilon^2\right) q\left(\sigma_x^2\right) q\left(a_\varepsilon\right) q\left(a_x\right) \prod_{j=1}^d q\left(b_j\right). \tag{8}$$

Typically, more restrictive density products facilitate the derivation of a variational algorithm but also negatively impact on the quality of the approximation. On the other hand, less severe restrictions may increase algebraic and computational complexity of the variational algorithms. The one used above provides a good trade-off between tractability and accuracy of the approximation.

The scope of MFVB and VMP is to provide expressions for the optimal q-densities that minimize the Kullback–Leibler divergence between the approximating densities themselves and the left-hand side of (8). The former is based upon a directed acyclic graph interpretation of the model, whereas the latter benefits from a factor graph representation and its subsequent division into factor graph fragments. While the two variational inference procedures lead to ostensibly different iterative algorithms, they converge to the identical posterior density function approximations in the case where the inputs of the algorithms are the same and parameters are updated in the same sequence, since they are each founded upon the same optimization problem (Wand, 2017). Details for fitting model (2) under restriction (8) via MFVB and VMP are shown in Sections 3.1 and 3.2.

3.1. Mean field variational Bayes

Mean field variational Bayes is a well established approximate Bayesian inference technique where tractability is achieved through factorization of the approximating density. Here we provide a short introduction to MFVB and we refer the reader to Section 3 of Wand et al. (2011) for fuller details on its derivation.

Let $\boldsymbol{z}$ be a vector of observed data and $\boldsymbol{\theta} \in \Theta$ represent all model parameters. The logarithm of the model marginal likelihood, $\log p(\boldsymbol{z})$, satisfies

$$\log p(\boldsymbol{z}) \geq \log \underline{p}(\boldsymbol{z}; q) \equiv \int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\boldsymbol{z}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \right\} d\boldsymbol{\theta},$$

where $\underline{p}(\boldsymbol{z}; q)$ is a lower-bound depending on a density function q defined over Θ and the joint density function $p(\boldsymbol{z}, \boldsymbol{\theta})$. It can be shown that maximizing the above lower-bound is equivalent to minimizing the Kullback-Leibler divergence between $q(\boldsymbol{\theta})$ and the joint posterior density function $p(\boldsymbol{\theta}|\boldsymbol{z})$,

$$\mathrm{KL}(q(\boldsymbol{\theta}) \| p(\boldsymbol{\theta}|\boldsymbol{z})) = \int q(\boldsymbol{\theta}) \log \left\{ \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{z})} \right\} d\boldsymbol{\theta}.$$

If the approximating density is factorized according to a partition $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_s)$ of $\boldsymbol{\theta}$ such that $q(\boldsymbol{\theta}) = \prod_{k=1}^s q(\boldsymbol{\theta}_k)$, as on the right side of (8), then the optimal approximating densities satisfy

$$q^*(\boldsymbol{\theta}_k) \propto \exp\left[E_{q(\boldsymbol{\theta} \backslash \boldsymbol{\theta}_k)}\{\log p(\boldsymbol{\theta}_k | \boldsymbol{z}, \boldsymbol{\theta} \backslash \boldsymbol{\theta}_k)\} \right], \quad k = 1, \ldots, s, \tag{9}$$

where $E_{q(\boldsymbol{\theta} \backslash \boldsymbol{\theta}_k)}$ denotes the expectation with respect to all the approximating densities except $q(\boldsymbol{\theta}_k)$ and $\boldsymbol{\theta} \backslash \boldsymbol{\theta}_k$ represents the entries of $\boldsymbol{\theta}$ with $\boldsymbol{\theta}_k$ omitted. Under mild regularity conditions it can be shown that optimization of the lower bound can be performed via a coordinate ascent scheme converging to a local maximizer.

Fig. 3 is a directed acyclic graph representation of model (2), which forms the basis for deriving an MFVB algorithm. The shaded circle corresponds to the observed data, the empty circles correspond to model parameters and auxiliary variables, and the small solid circles are used for hyperparameters. According to (9), the full conditional density functions of the nodes in the directed acyclic graph provide expressions for the optimal approximating densities. For instance, the full conditional density of $\boldsymbol{x}$ arising from model (2) and (3) is

$$p(\boldsymbol{x}|\text{rest}) \propto \exp\left[-\frac{1}{2} \left\{ \boldsymbol{x}^T \boldsymbol{L}^T \left(\frac{1}{\sigma_\varepsilon^2} \boldsymbol{K}^T \boldsymbol{K} + \frac{1}{\sigma_x^2} \operatorname{diag}(\boldsymbol{b}) \right) \boldsymbol{L} \boldsymbol{x} - \frac{2}{\sigma_\varepsilon^2} \boldsymbol{x}^T \boldsymbol{K}^T \boldsymbol{y} \right\} \right].$$

From application of (9) it follows that

$q^*(\boldsymbol{x})$ is a $N\left(\boldsymbol{\mu}_{q(\boldsymbol{x})}, \boldsymbol{\Sigma}_{q(\boldsymbol{x})}\right)$ density function $\qquad$ (10)

with

$$\boldsymbol{\mu}_{q(\boldsymbol{x})} \equiv \mu_{q(1/\sigma_\varepsilon^2)} \boldsymbol{\Sigma}_{q(\boldsymbol{x})} \boldsymbol{K}^T \boldsymbol{y} \text{ and } \boldsymbol{\Sigma}_{q(\boldsymbol{x})} \equiv \left(\mu_{q(1/\sigma_\varepsilon^2)} \boldsymbol{K}^T \boldsymbol{K} + \mu_{q(1/\sigma_x^2)} \boldsymbol{L}^T \operatorname{diag}(\mu_{q(\boldsymbol{b})}) \boldsymbol{L}\right)^{-1},$$

where $\mu_{q(1/\sigma_\varepsilon^2)}$, $\mu_{q(1/\sigma_x^2)}$ and $\mu_{q(\boldsymbol{b})}$ respectively denote the expectations of $1/\sigma_\varepsilon^2$, $1/\sigma_x^2$ and $\boldsymbol{b}$ computed with respect to the optimal approximating densities $q^*(\sigma_\varepsilon^2)$, $q^*(\sigma_x^2)$ and $q^*(\boldsymbol{b}) = \prod_{j=1}^d q^*\left(b_j\right)$ arising from the mean field restriction (8). Referring to restriction (8), the other MFVB approximations to the posterior density functions of the parameters and auxiliary variables depicted in the directed acyclic graph have the following optimal forms:

$q^*(b_j)$ is an Inverse-Gaussian$\left(\mu_{q(b_j)}, \lambda_{q(b_j)}\right)$ density function, for $j = 1, \dots, d$, $\qquad$ (11)

$q^*(\sigma_\varepsilon^2)$ is an Inverse-$\chi^2\left(\kappa_{q\left(\sigma_\varepsilon^2\right)}, \lambda_{q\left(\sigma_\varepsilon^2\right)}\right)$ density function, $\qquad$ (12)

$q^*(\sigma_x^2)$ is an Inverse-$\chi^2\left(\kappa_{q\left(\sigma_x^2\right)}, \lambda_{q\left(\sigma_x^2\right)}\right)$ density function, $\qquad$ (13)

$q^*(a_\varepsilon)$ is an Inverse-$\chi^2\left(\kappa_{q(a_\varepsilon)}, \lambda_{q(a_\varepsilon)}\right)$ density function $\qquad$ (14)

and $q^*(a_x)$ is an Inverse-$\chi^2\left(\kappa_{q(a_x)}, \lambda_{q(a_x)}\right)$ density function. $\qquad$ (15)

Details about the optimal density function parameters are given in the supplement. Such parameters can be obtained through the MFVB iterative scheme listed as Algorithm 1.

Algorithm 1 Mean field variational Bayes scheme for fitting the inverse problem model (2), using product density restriction (8).

Data Inputs: $\boldsymbol{y}$ ($m \times 1$), $\boldsymbol{K}$ ($m \times m$).
Hyperparameter Inputs: $A_\varepsilon > 0$, $A_x > 0$.
Initialize: $\mu_{q(1/\sigma_\varepsilon^2)} > 0$, $\mu_{q(1/\sigma_x^2)} > 0$, $\mu_{q(1/a_\varepsilon)} > 0$, $\mu_{q(1/a_x)} > 0$, $\boldsymbol{\mu}_{q(\boldsymbol{b})}$ ($d \times 1$) vector of positive elements.
$\kappa_{q(\sigma_\varepsilon^2)} \longleftarrow m+1$; $\kappa_{q(\sigma_x^2)} \longleftarrow d+1$; $\kappa_{q(a_\varepsilon)} \longleftarrow 2$; $\kappa_{q(a_x)} \longleftarrow 2$.
Cycle:

$\boldsymbol{\Sigma}_{q(\boldsymbol{x})} \longleftarrow \left(\mu_{q(1/\sigma_\varepsilon^2)} \boldsymbol{K}^T \boldsymbol{K} + \mu_{q(1/\sigma_x^2)} \boldsymbol{L}^T \operatorname{diag}(\boldsymbol{\mu}_{q(\boldsymbol{b})}) \boldsymbol{L}\right)^{-1}$
$\boldsymbol{\mu}_{q(\boldsymbol{x})} \longleftarrow \mu_{q(1/\sigma_\varepsilon^2)} \boldsymbol{\Sigma}_{q(\boldsymbol{x})} \boldsymbol{K}^T \boldsymbol{y}$
$\lambda_{q(\sigma_\varepsilon^2)} \longleftarrow \mu_{q(1/a_\varepsilon)} + \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\mu}_{q(\boldsymbol{x})}\|^2 + \operatorname{tr}\left(\boldsymbol{K}^T \boldsymbol{K} \boldsymbol{\Sigma}_{q(\boldsymbol{x})}\right)$
$\mu_{q(1/\sigma_\varepsilon^2)} \longleftarrow \kappa_{q(\sigma_\varepsilon^2)} / \lambda_{q(\sigma_\varepsilon^2)}$
$\lambda_{q(a_\varepsilon)} \longleftarrow \mu_{q(1/\sigma_\varepsilon^2)} + 1/A_\varepsilon^2$; $\mu_{q(1/a_\varepsilon)} \longleftarrow \kappa_{q(a_\varepsilon)} / \lambda_{q(a_\varepsilon)}$
$\boldsymbol{\tau}_1 \longleftarrow \left(\boldsymbol{L}\boldsymbol{\mu}_{q(\boldsymbol{x})}\right)^2 + \operatorname{diagonal}(\boldsymbol{L}\boldsymbol{\Sigma}_{q(\boldsymbol{x})}\boldsymbol{L}^T)$
$\lambda_{q(\sigma_x^2)} \longleftarrow \mu_{q(1/a_x)} + \boldsymbol{\mu}_{q(\boldsymbol{b})}^T \boldsymbol{\tau}_1$
$\mu_{q(1/\sigma_x^2)} \longleftarrow \kappa_{q(\sigma_x^2)} / \lambda_{q(\sigma_x^2)}$
$\lambda_{q(a_x)} \longleftarrow \mu_{q(1/\sigma_x^2)} + 1/A_x^2$; $\mu_{q(1/a_x)} \longleftarrow \kappa_{q(a_x)} / \lambda_{q(a_x)}$
$\boldsymbol{\tau}_2 \longleftarrow \mu_{q(1/\sigma_x^2)} \boldsymbol{\tau}_1$; $\boldsymbol{\mu}_{q(\boldsymbol{b})} \longleftarrow 1/\sqrt{\boldsymbol{\tau}_2}$
Output: $\boldsymbol{\mu}_{q(\boldsymbol{x})}, \boldsymbol{\Sigma}_{q(\boldsymbol{x})}, \boldsymbol{\mu}_{q(\boldsymbol{b})}, \lambda_{q(\boldsymbol{b})} \equiv \mathbf{1}_d, \kappa_{q(\sigma_\varepsilon^2)}, \lambda_{q(\sigma_\varepsilon^2)}, \kappa_{q(\sigma_x^2)}, \lambda_{q(\sigma_x^2)}, \kappa_{q(a_\varepsilon)}, \lambda_{q(a_\varepsilon)}, \kappa_{q(a_x)}, \lambda_{q(a_x)}.$

3.2. Variational message passing

The idea behind variational message passing as presented in Wand (2017) is that the same approximations in (9) can be achieved by exploiting a convenient factor graph representation of the model. A detailed description of VMP as a method for fitting statistical models that have a factor graph representation is provided in Sections 2–4 of Wand (2017) and briefly summarized here. The same notational conventions of Wand (2017) concerning message passing are used in this work.

A factor graph is an ensemble of factors connected to stochastic nodes by edges. Let f_h, $h = 1, \dots, r$, denote a generic factor and function of one or more stochastic nodes and θ_k, $k = 1, \dots, s$, be a generic stochastic variable represented by a node. If θ_k is a neighbour of f_h in the factor graph, then the messages passed from f_h to θ_k and from θ_k to f_h are functions of θ_k and are denoted by $\boldsymbol{m}_{f_h \to \theta_k}(\theta_k)$ and $\boldsymbol{m}_{\theta_k \to f_h}(\theta_k)$, respectively.

In the optic of deriving an algorithm, the VMP stochastic node to factor message updates are given by

$$\boldsymbol{m}_{\theta_k \to f_h}(\theta_k) \longleftarrow \propto \prod_{h' \neq h \,:\, k \in \operatorname{neighbours}(h')} \boldsymbol{m}_{f_{h'} \to \theta_k}(\theta_k) \qquad (16)$$

and the factor to stochastic node message updates are

$$\boldsymbol{m}_{f_h \to \theta_k}(\theta_k) \longleftarrow \propto \exp\left[E_{f_h \to \theta_k}\left\{\log f_h(\boldsymbol{\theta}_{\text{neighbours}})\right\}\right], \qquad (17)$$

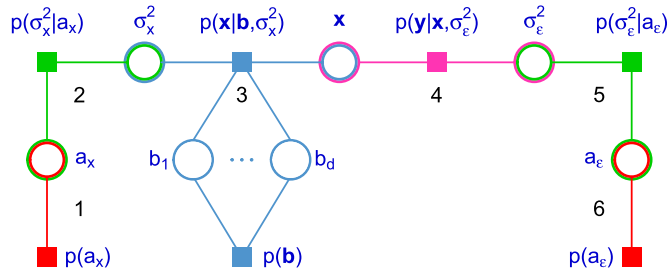where $E_{f_h \to \theta_k}$ denotes expectation with respect to the density function

Fig. 4. Factor graph representation of the Normal response model with Laplace penalization in (2), where the square nodes correspond to the density functions, or factors, at the right-hand side of (7). The circular nodes correspond to stochastic nodes of the q-density factorization in (8). The fragments are numbered 1 to 6, whereas colours identify different fragment types.

$$\frac{\prod_{k'\in \text{neighbours}(h)\setminus\{k\}} \boldsymbol{m}_{f_h\to\theta_{k'}}(\boldsymbol{\theta}_{k'})\boldsymbol{m}_{\theta_{k'}\to f_h}(\boldsymbol{\theta}_{k'})}{\prod_{k'\in \text{neighbours}\setminus\{k\}} \int \boldsymbol{m}_{f_h\to\theta_{k'}}(\boldsymbol{\theta}_{k'})\boldsymbol{m}_{\theta_{k'}\to f_h}(\boldsymbol{\theta}_{k'})d\boldsymbol{\theta}_{k'}}, \tag{18}$$

neighbours $\equiv \{k=1,\dots,s \,:\, \boldsymbol{\theta}_k \text{ is a neighbour of } f_h\}$ and the $\longleftarrow\!\!\propto$ symbol means that the function of $\boldsymbol{\theta}_k$ on the left-hand side is updated according to the expression on the right-hand side but that multiplicative factors not depending on $\boldsymbol{\theta}_k$ can be ignored. Upon convergence of the messages, the optimal q-densities are obtained via

$$q^*(\boldsymbol{\theta}_k) \propto \prod_{h\,:\,k\in\text{neighbours}} \boldsymbol{m}_{f_h\to\theta_k}(\boldsymbol{\theta}_k). \tag{19}$$

The messages above are typically proportional to an exponential family density function and so are such that

$$\boldsymbol{m}_{f_h\to\theta_k}(\boldsymbol{\theta}_k) \propto \exp\left\{\boldsymbol{T}(\boldsymbol{\theta}_k)^T\boldsymbol{\eta}_{f_h\to\theta_k}\right\} \quad\text{and}\quad \boldsymbol{m}_{\theta_k\to f_h}(\boldsymbol{\theta}_k) \propto \exp\left\{\boldsymbol{T}(\boldsymbol{\theta}_k)^T\boldsymbol{\eta}_{\theta_k\to f_h}\right\}, \tag{20}$$

where $\boldsymbol{T}(\boldsymbol{\theta}_k)$ is a sufficient statistic vector, and $\boldsymbol{\eta}_{f_h\to\theta_k}$ and $\boldsymbol{\eta}_{\theta_k\to f_h}$ are the message natural parameter vectors. Then for each parameter $\boldsymbol{\theta}_k$, the optimal approximating density $q^*(\boldsymbol{\theta}_k)$ belongs to an exponential family with natural parameter vector

$$\boldsymbol{\eta}_{q(\theta_k)} \equiv \prod_{h\,:\,k\in\text{neighbours}} \boldsymbol{\eta}_{f_h\to\theta_k}(\boldsymbol{\theta}_k) \tag{21}$$

that can be computed at convergence of a VMP algorithm.

The notion of a factor graph fragment, or simply *fragment*, allows for compartmentalization of algebra and computer code and can be exploited to fit inverse problems via VMP without resorting to calculations involving (16) and (17) each time a VMP algorithm is derived for a new model. The corresponding factor graph representation of (7) given the density product restriction (8) appears in Fig. 4. Colours mark different fragment types, in accordance with the nomenclature presented in Wand (2017) for variational message passing on factor graph fragments and numbers label seven factor graph fragments. Some of these have been studied in previous works. Those numbered 1, 2, 5 and 6 are already catalogued in Maestrini and Wand (2021) as *Inverse G-Wishart prior fragments* (1 and 6) and *iterated Inverse G-Wishart fragments* (2 and 5). Fragment number 4 corresponds to the *Gaussian likelihood fragment* treated in Section 4.1.5 of Wand (2017), whose notation can be aligned with that of model (2) by settings $\boldsymbol{A}$, $\boldsymbol{\theta}_1$ and θ_2 equal to the current $\boldsymbol{K}$, $\boldsymbol{x}$ and σ_ε^2, respectively. In the view of VMP, we can just read off from equations (38) and (39) of Wand (2017), which themselves originate from (17), and get the following updates for fragment 4 involving factor $p(\boldsymbol{y}\,|\,\boldsymbol{x},\sigma_\varepsilon^2)$ and stochastic nodes $\boldsymbol{x}$ and σ_ε^2:

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{x},\sigma_\varepsilon^2)\to\boldsymbol{x}} \longleftarrow \begin{bmatrix} \boldsymbol{K}^T\boldsymbol{y} \\ -\frac{1}{2}\,\text{vec}\left(\boldsymbol{K}^T\boldsymbol{K}\right) \end{bmatrix} \left\{\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{x},\sigma_\varepsilon^2)\leftrightarrow\sigma_\varepsilon^2}\right)_1 + 1\right\}\bigg/\left\{\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{x},\sigma_\varepsilon^2)\leftrightarrow\sigma_\varepsilon^2}\right)_2\right\}$$

and

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{x},\sigma_\varepsilon^2)\to\sigma_\varepsilon^2} \longleftarrow \begin{bmatrix} -n/2 \\ G_{\text{VMP}}\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{x},\sigma_\varepsilon^2)\leftrightarrow\boldsymbol{x}}; \boldsymbol{K}^T\boldsymbol{K}, \boldsymbol{K}^T\boldsymbol{y}, \boldsymbol{y}^T\boldsymbol{y}\right) \end{bmatrix},$$

where, according to the definition in Section 2.7 of Wand (2017), $G_{\text{VMP}}\left([\,\boldsymbol{v}_1^T\,\,\boldsymbol{v}_2^T\,]^T; \boldsymbol{Q}, \boldsymbol{r}, s\right) \equiv -(1/8)\text{tr}\left(\boldsymbol{Q}\{\text{vec}^{-1}(\boldsymbol{v}_2)\}^{-1}[\boldsymbol{v}_1\boldsymbol{v}_1^T \{\text{vec}^{-1}(\boldsymbol{v}_2)\}^{-1} - 2\boldsymbol{I}]\right) - (1/2)\boldsymbol{r}^T\{\text{vec}^{-1}(\boldsymbol{v}_2)\}^{-1}\boldsymbol{v}_1 - (1/2)s$, for a $d\times 1$ vector $\boldsymbol{v}_1$, $d^2\times 1$ vector $\boldsymbol{v}_2$ such that $\text{vec}^{-1}(\boldsymbol{v}_2)$ is symmetric, $d\times d$ matrix $\boldsymbol{Q}$, $d\times 1$ vector $\boldsymbol{r}$ and $s\in\mathbb{R}$.

New calculations are needed only for fragment 3 to compose a VMP algorithm for the whole model. In fragment 3, factor $p(\boldsymbol{b})$ symbolizes the specification

$$b_j \overset{\text{ind.}}{\sim} \text{Inverse-}\chi^2(2,1), \quad j=1,\dots,d, \tag{22}$$

but may also represent other penalizations. Examples of alternative shrinkage distributions are displayed in Table 1 and discussed in Section 3.3. The factor $p\left(x|b,\sigma_x^2\right)$ of fragment 3 corresponds to the possibly improper specification (3). The logarithm of this likelihood factor is

$$\log p\left(x|b,\sigma_x^2\right) = -\frac{d}{2}\log\left(\sigma_x^2\right) - \frac{1}{2\sigma_x^2}x^T L^T \operatorname{diag}\left(b\right) Lx + \frac{1}{2}\sum_{i=1}^{d}\log\left(b_i\right) + \text{const.} \tag{23}$$

Combining (23) with the auxiliary variable distributional assumption in (22) we get the VMP scheme listed as Algorithm 2 for fitting the penalization part of model (2). This originates from first noticing that, as a function of x, (23) can be written as

$$\log p\left(x|b,\sigma_x^2\right) = \frac{1}{\sigma_x^2}\left[\begin{array}{c} x \\ \operatorname{vec}\left(xx^T\right) \end{array}\right]^T \left[\begin{array}{c} 0 \\ -\frac{1}{2}\operatorname{vec}\left(L^T \operatorname{diag}\left(b\right) L\right) \end{array}\right] + \text{const,}$$

which in light of (17) and (20) originates the message sent from factor $\left(x|b,\sigma_x^2\right)$ to node x

$$m_{p(x|b,\sigma_x^2)\to x}\left(x\right) = \exp\left\{\left[\begin{array}{c} x \\ \operatorname{vec}\left(xx^T\right) \end{array}\right]^T \boldsymbol{\eta}_{p(x|b,\sigma_x^2)\to x}\right\}.$$

This message is within the Multivariate Normal family, with

$$\boldsymbol{\eta}_{p(x|b,\sigma_x^2)\to x} \equiv E_{\boxtimes}\left(1/\sigma_x^2\right)\left[\begin{array}{c} 0 \\ -\frac{1}{2}\operatorname{vec}\left(L^T \operatorname{diag}\left\{E_{\oplus}\left(b\right)\right\} L\right) \end{array}\right],$$

where, following (18), $E_{\boxtimes}$ denotes expectation with respect to the normalization of

$$m_{p(x|b,\sigma_x^2)\to\sigma_x^2}\left(\sigma_x^2\right) m_{\sigma_x^2\to p(x|b,\sigma_x^2)}\left(\sigma_x^2\right)$$

and $E_{\oplus}$ denotes expectation with respect to the normalization of

$$m_{p(x|b,\sigma_x^2)\to b}\left(b\right) m_{b\to p(x|b,\sigma_x^2)}\left(b\right).$$

Rewriting $\log p\left(x|b,\sigma_x^2\right)$ as a function of σ_x^2 and b and applying similar considerations we get to Algorithm 2 (see the supplement for full derivations).

Algorithm 2 Variational message passing inputs, updates and outputs of the penalization likelihood fragment given by (3) and (22), and corresponding to factor graph fragment 3 of Fig. 4.

Inputs: $\boldsymbol{\eta}_{p(x|b,\sigma_x^2)\to x}, \boldsymbol{\eta}_{x\to p(x|b,\sigma_x^2)}, \boldsymbol{\eta}_{p(x|b,\sigma_x^2)\to\sigma_x^2}, \boldsymbol{\eta}_{\sigma_x^2\to p(x|b,\sigma_x^2)}.$
Updates:

$\mu_{q(1/\sigma_x^2)} \longleftarrow \left\{\left(\boldsymbol{\eta}_{p(x|b,\sigma_x^2)\leftrightarrow\sigma_x^2}\right)_1 + 1\right\}\Big/\left\{\left(\boldsymbol{\eta}_{p(x|b,\sigma_x^2)\leftrightarrow\sigma_x^2}\right)_2\right\}$

$\boldsymbol{\Omega}_1 \longleftarrow -\frac{1}{2}\left\{\operatorname{vec}^{-1}\left(\left(\boldsymbol{\eta}_{p(x|b,\sigma_x^2)\leftrightarrow x}\right)_2\right)\right\}^{-1}$

$\boldsymbol{\omega}_2 \longleftarrow \boldsymbol{\Omega}_1\left(\boldsymbol{\eta}_{p(x|b,\sigma_x^2)\leftrightarrow x}\right)_1$; $\boldsymbol{\omega}_3 \longleftarrow L\boldsymbol{\omega}_2$; $\boldsymbol{\omega}_4 \longleftarrow \operatorname{diagonal}\left(L\boldsymbol{\Omega}_1 L^T\right)$

$\boldsymbol{\omega}_5 \longleftarrow \boldsymbol{\omega}_3^2 + \boldsymbol{\omega}_4$; $\boldsymbol{\omega}_6 \longleftarrow \mu_{q(1/\sigma_x^2)}\boldsymbol{\omega}_5$

$\mu_{q(b)} \longleftarrow 1/\sqrt{\boldsymbol{\omega}_6}$; $\boldsymbol{\Omega}_7 \longleftarrow L^T \operatorname{diag}\left(\mu_{q(b)}\right) L$
until convergence.

Outputs: $\boldsymbol{\eta}_{p(x|b,\sigma_x^2)\to x} \longleftarrow \mu_{q(1/\sigma_x^2)}\left[\begin{array}{c} 0 \\ -\frac{1}{2}\operatorname{vec}\left(\boldsymbol{\Omega}_7\right) \end{array}\right]$; $\boldsymbol{\eta}_{p(x|b,\sigma_x^2)\to\sigma_x^2} \longleftarrow -\frac{1}{2}\left[\begin{array}{c} m_L \\ \mu_{q(b)}^T\boldsymbol{\omega}_5 \end{array}\right].$

The combination of Algorithm 2 with the Inverse G-Wishart fragment algorithms of Maestrini and Wand (2021) and the Gaussian likelihood fragment algorithm of Wand (2017) gives rise to a full VMP procedure for fitting and approximate inference on model (2). Complete details about the implementation of VMP for fitting the inverse problem model (2) are provided in the supplement. At convergence of the VMP procedure, (19) and (21) can be applied to obtain the approximating densities in explicit form. For instance, we have

$$q^*(x) \propto \exp\left\{\left[\begin{array}{c} x \\ \operatorname{vec}\left(xx^T\right) \end{array}\right]^T \boldsymbol{\eta}_{q(x)}\right\}, \text{ with } \boldsymbol{\eta}_{q(x)} \equiv \boldsymbol{\eta}_{p(x|b,\sigma_x^2)\to x} + \boldsymbol{\eta}_{p(y|x,\sigma_\varepsilon^2)\to x},$$

which can be rewritten in terms of the common parameters of the approximating density (10) of MFVB by setting

$$\boldsymbol{\mu}_{q(x)} = \boldsymbol{\Sigma}_{q(x)}\left(\boldsymbol{\eta}_{q(x)}\right)_{1:m} \quad \text{and} \quad \boldsymbol{\Sigma}_{q(x)} = -\frac{1}{2}\operatorname{vec}^{-1}\left\{\left(\boldsymbol{\eta}_{q(x)}\right)_{(m+1):m^2}\right\}.$$

Both MFVB and VMP can achieve the same approximations (10)–(15) since it is possible to establish a direct correspondence between the initialisation values of the two variational procedures. For example, using the hyperparameter input A_x in Algorithm 1 for

Table 1

Distributions of the auxiliary variables $b_j > 0$, $j = 1 \ldots d$, that produce the penalizations analyzed in Neville et al. (2014) when introduced in model (2) in lieu of the Laplace penalization. Here the shape parameter $\lambda > 0$ is fixed. For each distribution, the MFVB or VMP update for $\boldsymbol{\mu}_{q(b)}$ is displayed in the last column. When Algorithm 1 is used to perform MFVB, $\zeta \equiv \tau_2$; when Algorithm 2 is used for running VMP, $\zeta \equiv \boldsymbol{\omega}_6$.

Penalization	Density function	MFVB or VMP update
Horseshoe	$p(b_j) = \pi^{-1} b_j^{-1/2} (1 + b_j)^{-1}$	$\boldsymbol{\mu}_{q(b)} \longleftarrow \dfrac{2}{\zeta \odot \exp\left(\frac{1}{2}\zeta\right) \odot E_1\left(\frac{1}{2}\zeta\right)} - 1$
Negative-Exponential-Gamma	$p(b_j) = \lambda b_j^{\lambda-1} (1 + b_j)^{-\lambda-1}$	$\boldsymbol{\mu}_{q(b)} \longleftarrow (2\lambda + 1)\mathcal{R}_{2\lambda}\left(\sqrt{\zeta}\right) \odot \sqrt{\zeta}$
Generalized Double Pareto	$p(b_j) = \frac{1}{2}(1 + \lambda)\lambda^{1+\lambda}$ $\times b_j^{(\lambda-2)/2} e^{\lambda^2 b_j/4} \mathcal{D}_{-\lambda-2}\left(\lambda\sqrt{b_j}\right)$	$\boldsymbol{\mu}_{q(b)} \longleftarrow \dfrac{\sqrt{2}(\lambda+1)}{\sqrt{\zeta} \odot (\sqrt{2\lambda+\zeta})}$

MFVB corresponds to a call of VMP to the Inverse G-Wishart Prior Fragment (Maestrini and Wand, 2021, Algorithm 1) with inputs $G_\Theta = G_{\text{diag}}$, $\xi_\Theta = 1$, $\Lambda_\Theta = A_x^{-2}$ and G_{diag} being a graph corresponding to a diagonal matrix, to initialise the natural parameter vector $\boldsymbol{\eta}_{p(a_x) \to a_x}$ of the message sent from factor $p(a_x)$ to stochastic node a_x. Further details about the initialisation of VMP are provided in the supplement.

3.3. Alternative response and penalization distributions

The message passing on factor graph fragments paradigm allows for flexible imposition of non-Normal response distributions. For instance, fragment 4 of Fig. 4 can be replaced by one of the likelihood fragments identified in Nolan and Wand (2017), Maestrini and Wand (2018) or McLean and Wand (2019) to accommodate a variety of response distributions such as, for instance, binary-logistic, Poisson, Negative Binomial, t, Asymmetric Laplace, Skew Normal, Skew t and Finite Normal Mixtures.

Other penalization structures can be easily incorporated by varying the distributional assumption on the vector of auxiliary variables $\boldsymbol{b}$. Neville et al. (2014) studied MFVB inference for three continuous sparse signal shrinkage distributions, namely the Horseshoe, Negative-Exponential-Gamma and Generalized Double Pareto distributions, that can replace the Laplace penalization employed in model (2). References for the development of these sparse shrinkage priors are respectively Carvalho et al. (2010), Griffin and Brown (2011) and Armagan et al. (2013).

The derivation of variational algorithms for models containing these shrinkage distributions can be quite challenging. In fact, the variational algorithm algebraic complexity and inference performance rely upon an accurate choice of their auxiliary variable representations. Neville et al. (2014) propose two alternative auxiliary variable representations for each of the aforementioned shrinkage distributions by making use of either one or two sets of auxiliary variables. Their empirical and theoretical evidence show the supremacy of representations based on a single set of auxiliary variables in terms of posterior density approximation and computational complexity for all three cases. If this auxiliary variable representation is chosen, then the three penalizations can be easily imposed in model (2) as a replacement to the Laplace distribution by simply modifying the distributional assumption on the auxiliary vector $\boldsymbol{b}$. Algorithms 1 and 2 can still be used by simply replacing the update for $\boldsymbol{\mu}_{q(b)}$ with one of those listed in the last column of Table 1. Some expressions in Table 1 related to the Negative-Exponential-Gamma and Generalized Double Pareto cases depend on the parabolic cylinder function of order v, $\mathcal{D}_v(x)$, and $\mathcal{R}_v(x) \equiv \mathcal{D}_{-v-2}(x)/\mathcal{D}_{-v-1}(x)$, for $v > 0$ and $x > 0$. Efficient computation of function $\mathcal{R}_v(x)$ is discussed in Appendix A of Neville et al. (2014).

Several other penalizations can be imposed on the base inverse problem model. The penalization in model (2) is a particular case of the Bayesian lasso of Park and Casella (2008) that makes use of a Gamma prior on the Laplace squared scale parameter. Tung et al. (2019) show the use of MFVB for variable selection in generalized linear mixed models via Bayesian adaptive lasso. Ormerod et al. (2017) develop a MFVB approximation to a linear model with a spike-and-slab prior on the regression coefficients. A detailed discussion on variable selection priors and variational inference fitting goes beyond the scope of this article. However, for the analysis of archaeological data provided in the supplement we show how the Laplace penalization can be easily replaced by a Horseshoe prior without deriving a VMP algorithm from scratch. In the same real data analysis we replace the Normal response assumption with a Skew Normal one.

4. Streamlined variational algorithm updates

In typical inverse problem applications the number of observations can be very high and a naïve implementation of Algorithms 1 and 2 may lead to a bottleneck due to operations involving a large contrast matrix $\boldsymbol{L}$ and big matrix inversion steps related to $\boldsymbol{K}$. However, the structure of matrices $\boldsymbol{L}$ and $\boldsymbol{K}$ is such that computationally expensive algorithm updates may be efficiently performed. In this section we propose solutions to simplify algorithm updates and reduce their computational complexity. The results shown here are designed for one- and two-dimensional problems but are applicable to extensions to higher dimensions.

4.1. Removal of the contrast matrix

The contrast matrix L is a potentially massive sparse matrix that one does not want to compute or store. The number of rows of L is given by the d differences between the elements of x considered, whereas the number of columns is given by the length m of x. For one-dimensional problems with a first nearest neighbour structure only the two longest diagonals of L have non-zero elements, as shown in (4). The contrast matrix of two-dimensional problems under the same assumptions is sparse and has number of non-zero elements equal to twice the number of differences between elements of x, that is $2d$.

The updates of the variational Algorithms 1 and 2 that make use of matrix L have the following forms:

$$L v, \text{ where } v \text{ is a vector;} \tag{24}$$

$$\text{diagonal}(L M L^T), \text{ where } M \text{ is a symmetric positive definite matrix;} \tag{25}$$

$$L^T \text{diag}(w) L, \text{ where } w \text{ is a vector.} \tag{26}$$

The results presented in the supplementary material allow efficient computation of the MFVB and VMP updates that utilise matrix L in the forms described in (24)–(26). One- and two-dimensional problems are considered and the results can be potentially extended to higher dimensions, e.g. for voxel-type data in three-dimensions.

4.2. Sparsification of K

The matrix K is a potentially big matrix whose size depends on the number of observations and the length of x. It is easy to notice from the MFVB scheme presented as Algorithm 1 that the algebraic operations where matrix K appears have the following generic forms:

$$\left(s_1 K^T K + s_2 L^T \text{diag}(\omega) L\right)^{-1}, \text{ where } s_1 \text{ and } s_2 \text{ are scalar positive numbers;} \tag{27}$$

$$M K^T v, \text{ where } M \text{ is a symmetric positive definite matrix and } v \text{ is a vector;} \tag{28}$$

$$\|u - K t\|^2, \text{ where } u \text{ and } t \text{ are vectors;} \tag{29}$$

$$\text{tr}(K^T K N), \text{ where } N \text{ is a matrix.} \tag{30}$$

For what concerns VMP, the expression in (27) appears in the update for Ω_1 of Algorithm 2, whereas those in (28)–(30) arise in the Gaussian likelihood fragment numbered as fragment 5 in the Fig. 4 factor graph. Visual inspection of (27)–(30) suggests it is worth studying the structure of K and $K^T K$ for a computationally efficient implementation of the variational algorithm updates.

The focus of this section is placed on two-dimensional inverse problems. Again, we restrict our attention to the case where X and Y are both $m_1 \times m_2$ matrices and each element of X has a one-to-one correspondence with an element having the same position in Y. Under these conditions K is a square matrix of size $m \times m$, with $m = m_1 m_2$. If model (2) is used, setting $x = \text{vec}(X)$ and $y = \text{vec}(Y)$, the matrix K has the following structure:

$$K = \begin{bmatrix} K_1 & K_2 & \cdots & K_{m_2} \\ K_2 & K_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & K_2 \\ K_{m_2} & \cdots & K_2 & K_1 \end{bmatrix}, \text{ with } K_i = \begin{bmatrix} K_{i,1} & K_{i,2} & \cdots & K_{i,m_1} \\ K_{i,2} & K_{i,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & K_{i,2} \\ K_{i,m_1} & \cdots & K_{i,2} & K_{i,1} \end{bmatrix}, i = 1, \ldots, m_2.$$

Therefore K is a symmetric block-Toeplitz matrix with m_2 unique sub-blocks, each being $m_1 \times m_1$ symmetric Toeplitz matrices. For simple unidimensional problems K is a symmetric Toeplitz matrix.

Both the MFVB and VMP updates

$$\Sigma_{q(x)} \longleftarrow \left(\mu_{q(1/\sigma_\varepsilon^2)} K^T K + \mu_{q(1/\sigma_x^2)} L^T \text{diag}(\mu_{q(b)}) L\right)^{-1} \text{ of Algorithm 1 and} \tag{31}$$

$$\Omega_1 \longleftarrow -\frac{1}{2} \left\{ \text{vec}^{-1}\left(\left(\eta_{p(x|b,\sigma_x^2) \leftrightarrow x}\right)_2\right) \right\}^{-1} \text{ of Algorithm 2} \tag{32}$$

require inversion of a matrix of size $m \times m$. From (31) it is easy to notice that the structure of the matrix being inverted is influenced by K through $K^T K$. A possible idea to reduce computational burden induced by these updates is to sparsify K in such a way that also $K^T K$ and the final matrix to invert are sparse. Since K linearly links elements in X with those in Y and given the one-to-one correspondence between X and Y, it is reasonable to set to zero the matrix K elements which correspond to interactions between pairs of locations whose distance exceeds a certain *truncation* value $\ell \in \mathbb{N}$, with $\ell < \min(m_1, m_2)$. This strategy can find application, for instance, both in the context of biomedical imaging discussed in Section 5 and archaeological field survey treated in the supplementary material. More formally, this consists in setting to zero the entries of K that model dependence between pairs of elements of X and Y, $(X_{ij}, Y_{i'j'})$, such that
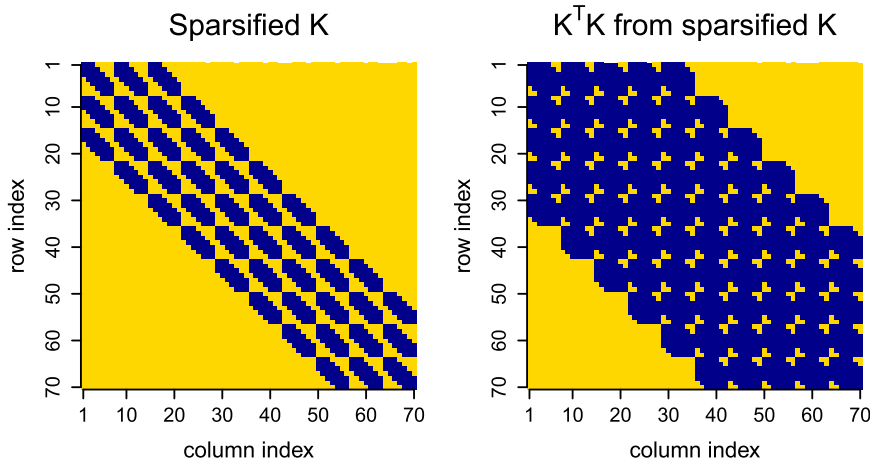
$$\max\left(|i - i'|, |j - j'|\right) > \ell.$$

Fig. 5. Representation of a sparse K matrix of size 70×70 for an inverse problem over a pixel grid of size 7×10 (left panel) and the corresponding $K^T K$ matrix (right panel). These matrices are obtained by using a truncation number $\ell = 2$. Non-zero values are depicted in blue.

The resultant K is an ℓ-block-banded matrix whose sub-blocks are ℓ-banded matrices. Also $K^T K$ may result in a sparse matrix for particular choices of ℓ, as stated in the following result.

Result 2. *Let A be an ℓ-block-banded matrix of size $m \times m$ with ℓ-banded sub-blocks and such that $0 < \ell < (m-1)/2$. Then $A^T A$ is a symmetric 2ℓ-block-banded matrix whose sub-blocks are 2ℓ-banded matrices.*

Fig. 5 depicts the K matrix and $K^T K$ block for an inverse problem where X and Y have size 7×10 and K is sparsified by applying a truncation value $\ell = 2$. The blue colour indicates non-zero matrix entries. In this case, matrix K is 2-block-banded matrix with 2-banded sub-blocks, whereas $K^T K$ is a 4-block-banded matrix with 4-banded sub-blocks.

It is easy to check through Lemma 6 by applying such a sparsification strategy to $K^T K$ that the same sparsity structure is imposed to the matrices being inverted in updates (31) and (32). Hence, for appropriate choices of ℓ, the updates involve inversion of a sparse matrix having a block-banded structure and banded matrices in the main block-diagonals.

The suggested sparsification strategy has a physical interpretation. In the two-dimensional problems under examination each element of Y linearly depends on a subset of the elements of X through K. If the elements of K are set to zero according to a truncation number ℓ, then such subset is given by those elements of X that fall inside of a circle of diameter $2\ell + 1$ around the element of X having one-to-one correspondence with the Y entry. Sparsifying the expression to be inverted in (31) means setting to zero some elements of the precision matrix $\Omega_{q(x)} = \Sigma_{q(x)}^{-1}$, where $\Sigma_{q(x)}$ is the covariance matrix of the optimal approximating density function $q^*(x)$ in (10). Since $q^*(x)$ is a Multivariate Normal density function, $\left(\Omega_{q(x)}\right)_{ij} = 0$, for $i, j = 1, \ldots, m$, if and only if x_i and x_j are conditionally independent given all the other elements of x.

Note that differently from the algebraic results proposed for removal of L, the sparsification applied to K comes from nullifying interactions between elements of X and Y, and therefore it introduces another level of approximation to the variational fitting procedure.

4.2.1. Block-banded matrix algebra

Asif and Moura (2005) propose two algorithms to invert block-banded matrices whose inverses are positive definite matrices: one resorting to Cholesky factors and an alternative implementation that avoids Cholesky factorizations. These algorithms require the inversion of smaller matrices having the size of the block-banded matrix sub-blocks. In the two-dimensional inverse problems under examination these sub-blocks have a banded structure. An approach for inverting banded matrices is described in Kılıç and Stanica (2013). These algebraic approaches for handling block-banded and banded matrices may allow for stable computation of variational algorithm steps involving sparse matrix inversions such as (27). However, the simplest way to perform efficient sparse matrix inversion and computations is to employ software for sparse matrix algebra. The functions contained in the R package `spam` (Furrer and Sain, 2010) allow efficient management of sparse matrices and implement matrix operations. This package can be used in combination with package `spam64` Gerber et al. (2017) to speed up such functions in 64 bit machines. Well established software is also available for lower-level languages such as the linear algebra libraries Armadillo and Eigen for C++ coding.

In general, matrices having banded or block-banded inverses are full matrices. Nonetheless, the inverse of a banded matrix may be referred to as *band-dominated matrix* (Bickel and Lindner, 2012), since the entries of its inverse exponentially decay with the distance from the main diagonal (Demko et al., 1984, Theorem 2.4). This property can be generalized to block-banded matrices and the inverse of a block-banded matrix can be approximated by a block-banded matrix with the same sparsity structure, i.e. with zero blocks off the main block band (Wijewardhana and Codreanu, 2016). The blocks outside the ℓ-block band of a symmetric positive definite matrix having an ℓ-block-banded inverse are called *nonsignificant* blocks and those in the ℓ-block band are called *significant* blocks. Theorem

3 of Asif and Moura (2005) states that nonsignificant blocks can be obtained from the significant ones. Then, a possible way to further speed-up algebra involving these matrices and reduce memory usage is to impose a block-banded structure to the precision matrix $\boldsymbol{\Omega}_{q(\boldsymbol{x})}$ and approximate $\boldsymbol{\Sigma}_{q(\boldsymbol{x})}$ with a block-banded matrix having the same structure of $\boldsymbol{\Omega}_{q(\boldsymbol{x})}$. In this case only the significant blocks of the covariance matrix $\boldsymbol{\Sigma}_{q(\boldsymbol{x})}$, which solely depend on the significant blocks of $\boldsymbol{\Omega}_{q(\boldsymbol{x})}$, need to be computed.

5. Biomedical data study

This section demonstrates the use of variational inference for two-dimensional inverse problems motivated by a biomedical application. An illustration on a real dataset and on simulations that mimic the real data are provided.

We assess the performances of variational inference through comparison with MCMC and computation of accuracy. For a generic univariate parameter θ, the approximation accuracy of a density $q(\theta)$ to a posterior density $p(\theta|\boldsymbol{y})$ is measured through

$$\text{accuracy} \equiv 100 \left\{ 1 - \frac{1}{2} \int_{-\infty}^{\infty} \left| q(\theta) - p(\theta|\boldsymbol{y}) \right| d\theta \right\}\%, \tag{33}$$

so that $0\% \leq \text{accuracy} \leq 100\%$, with 100% indicating perfect matching between the approximating and posterior density functions. We compute accuracy using Markov chain Monte Carlo as a benchmark. A standard Metropolis–Hastings algorithm Metropolis et al. (1953); Hastings (1970) is used to produce approximate samples from the posterior distribution. The Markov chains are started at feasible points in the parameter space and the retained samples are used to approximate the corresponding posterior density functions via kernel density estimation. Accuracy is then obtained from (33) with replacement of $p(\theta|\boldsymbol{y})$ by MCMC density estimates of the posterior density functions. Variational inference is performed removing the contrast matrix $\boldsymbol{L}$ through Lemmas 4–6 and sparsifying the matrix $\boldsymbol{K}$ via truncation of interactions, as explained in Section 4. Also our MCMC implementation does not make direct use of matrices $\boldsymbol{L}$ and $\boldsymbol{K}$ to reduce computational burden. The simulation study was run on a personal computer with a 64 bit Windows 10 operating system, an Intel i7-7500U central processing unit at 2.7 gigahertz and 16 gigabytes of random access memory. Variational inference was fully performed in R, whereas MCMC was run in R with subroutines replacing $\boldsymbol{L}$ and $\boldsymbol{K}$ matrix operations implemented in C++.

5.1. Real biomedical data

We test the performance of our variational inference approach on a real biomedical application from the realm of tomographic data. Tomography aims to display cross-sections through human and animal bodies, or other solid objects, using data collected around the body.

The data, kindly provided by BioEmission Technology Solutions, Athens, Greece, were collected to illustrate a small animal imaging system, gamma-eye, which can be used in biotechnology and pre-clinical medical research (Georgiou et al., 2017). A technetium radioisotope (99m Tc-MIBI) was injected via the tail vein of a mouse and mainly absorbed by organs such as heart, liver and kidneys, and then excreted. In humans such techniques are used to monitor heart function and mice are often used in pre-clinical studies. A single plane gamma-camera image of an adult mouse was collected with the camera at a distance of 5 mm from the nearest point of the mouse and 35 mm from the support bed. The 29×58, pixels of the data image are 1.7 mm apart giving a field of view of about 5 cm by 10 cm. The mouse was anesthetized and so this corresponds to the "at rest" part of a human scan which would also involve a "stress test". The total data recording time was 3 hours.

The objective is to reconstruct an image by removing blur from the observed scan of the mouse shown and denoted as $\boldsymbol{Y}$ in Fig. 6. We adopt model (2) setting $\boldsymbol{y} = \text{vec}(\boldsymbol{Y})$ and using a Gaussian kernel matrix $\boldsymbol{K}$ obtained through (6) with $\delta = 0.7$. Hence, $\boldsymbol{y}$ and $\boldsymbol{x}$ are vectors of length equal to the number of pixels, 1,682, and $\boldsymbol{K}$ has size $1,682 \times 1,682$. We set to zero the elements of $\boldsymbol{K}$ expressing interactions between locations that have 10 or more pixels between each other using $\ell = 10$. The random walk Metropolis–Hastings algorithm is used to produce approximate samples from the posterior distribution by simulating a Markov chain with burn-in of 1000 followed by 100,000, then thinned by a factor of 20. Hyperparameters are set to values that give rise to diffuse priors. Specifically, $A_\varepsilon = A_x = 10^5$. Estimates of $\boldsymbol{X}$ obtained via variational inference and MCMC are included in Fig. 6. Here the term variational inference refers to both MFVB and VMP, as they both provide the same results by construction. The estimate of $\boldsymbol{X}$ obtained via variational inference corresponds to the inverse vectorization of $\boldsymbol{\mu}_{q(\boldsymbol{x})}$ from (10), whereas that of MCMC is given by the mean of the sampled chains.

Fig. 6 also displays approximate posterior density plots for a selection of six representative pixels. Five of the six selected pixels correspond to targeted organs of the mouse, namely thyroid, liver, kidneys and bladder, while the remaining pixel is located outside but near the mouse body. Overall variational approximations provide good image reconstruction and facilitate visualisation of the mouse body shape and organs. The posterior density approximations are also satisfactory in terms of accuracy, that is, area overlap between MCMC. As typical of variational approximations based on mean field restrictions, variational inference underestimates the variance of the approximate posterior densities. Plots of some representative bivariate posterior densities are also provided in the supplement. These show that in all cases the variational approximation covers a smaller area, indicating that it underestimates uncertainty relative to MCMC.
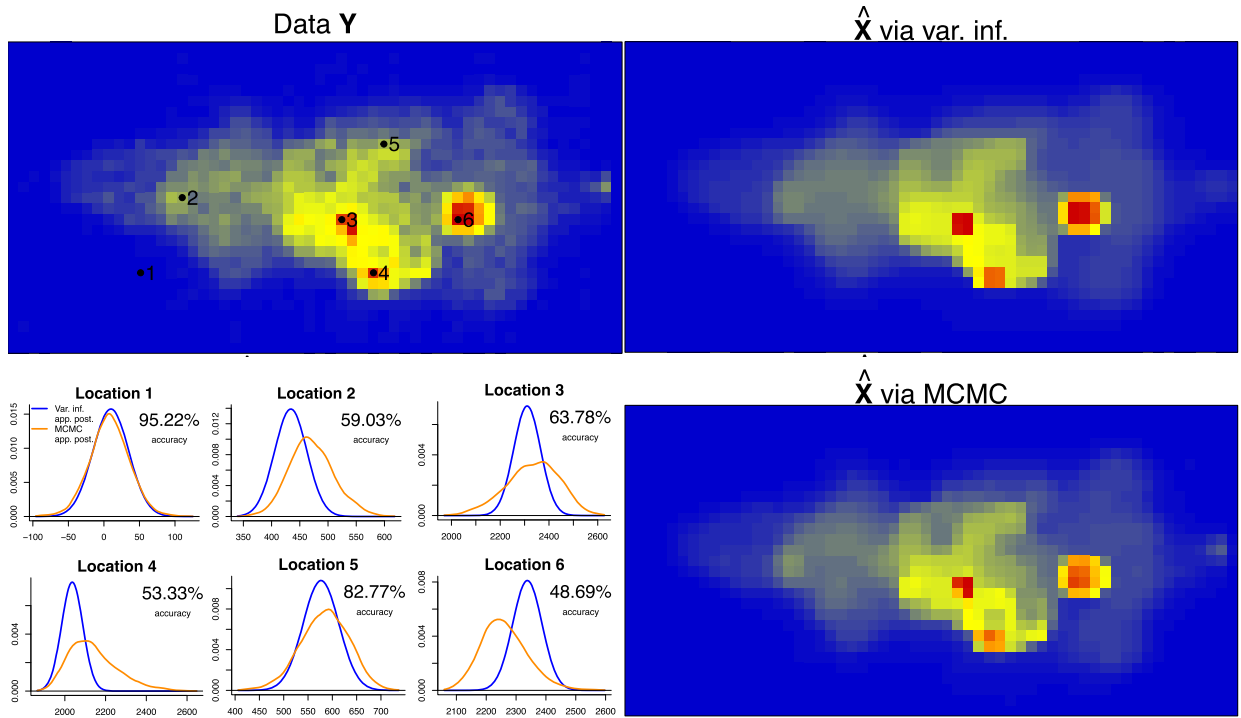
Fig. 6. Analysis of the data provided by BioEmission Technology Solutions, Athens, Greece, performed by fitting model (2) via variational inference and MCMC. The panel on the upper-left side displays the observed data $\boldsymbol{Y}$. The two panels on the right-hand side show, from top to bottom, the reconstruction $\widehat{\boldsymbol{X}}$ of $\boldsymbol{X}$ obtained via variational inference and MCMC. The plots on the lower-left side show the approximate marginal posterior densities produced through variational inference (blue lines) and MCMC (orange lines) for a selection of pixels.

5.2. Simulated biomedical data

We employ the real biomedical data image processed through MCMC, the one corresponding to the lower-right panel of Fig. 6, to simulate datasets and study the performance of variational inference in comparison with MCMC. In this simulation study we also keep track of computational times and calculate percentages of coverage. For a given parameter, the percentage of coverage corresponds to the proportion of simulations where the true parameter falls inside its 95% credible interval obtained through variational inference.

Let $\widehat{\boldsymbol{X}}_{\mathrm{MCMC}}$ be the inverse vectorization of the estimate of $\boldsymbol{x}$ obtained as the sample mean of the corresponding MCMC chains. We simulate data through:

$$\boldsymbol{y} = \boldsymbol{K}\,\mathrm{vec}(\widehat{\boldsymbol{X}}_{\mathrm{MCMC}}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma_\varepsilon^2 \boldsymbol{I}),$$

with $\sigma_\varepsilon = 50$ and $\boldsymbol{K}$ generated according to (6), using δ values from the set $(0.7, 0.8, 0.9)$ and without truncating interactions ($\ell = \infty$). The example plots provided in the supplement show how the blur increases for higher δ. For each δ value we generate 100 datasets and fit model (2) via both MFVB and MCMC. For fitting we apply the same $\boldsymbol{K}$ matrix used to generate the datasets, i.e. without truncation of interactions, but also fit the model again setting to zero the elements of $\boldsymbol{K}$ associated with pairs of observations whose distance exceeds a truncation value $\ell = 5$. The MFVB algorithm is stopped when the relative difference of estimates of $\boldsymbol{X}$ goes below 10^{-2} between two iterations. We generate Markov chains of length 6,000 and retain 5,000 of them after discarding 1,000 warm-up samples.

Table 2 summarizes the results of the simulation study including: accuracy of the variational inference estimates of $\boldsymbol{X}$ versus MCMC; variational inference percentage of coverage for $\boldsymbol{X}$, i.e. the number of times the entries of $\widehat{\boldsymbol{X}}_{\mathrm{MCMC}}$ falls inside their 95% variational inference credible intervals; variational inference and MCMC computational times. Average and standard deviations (in brackets) of each indicator are displayed for each combination of δ and ℓ values.

For each pixel of each simulated dataset we compute the accuracy of the variational approximation using (33) and then average over the 100 replicates. We then calculate average and standard deviation over all the entries of $\boldsymbol{X}$ and display these in Table 2. We repeat the same procedure for measuring the percentage of coverage performances. The mean accuracy values range between 88.07 and 87.10, whereas the mean percentages of coverage are between 95.09 and 92.75 and therefore close to the nominal 95% level. Both accuracy and coverage performances slightly degrade for higher values of δ and blur.

The variational inference and MCMC computational times are displayed in minutes and show that variational inference is around 100 times faster for the $\delta = 0.7$ setting. Imposing a truncation $\ell = 5$ to $\boldsymbol{K}$ reduces the MCMC computational times by about six to seven minutes on average for the three δ values under examination as opposed to not applying truncation. The time performances of variational inference are not particularly affected by truncation as the R package spam efficiently manages the algorithm updates

Table 2

Results of the simulation study based on 100 replicates per δ value generated using the kernel matrix K defined in (6) without truncation. Fitting is performed via variational inference and MCMC using the same K matrix employed to generate the data (no truncation) and also imposing a truncation threshold $\ell = 5$ to the interactions expressed by K. Higher δ values correspond to more blur. The smaller ℓ, the sparser the K matrix used to fit the model is.

		$\ell = \infty$ (no truncation)	$\ell = 5$
$\delta = 0.7$	Var. inf. vs MCMC accuracy for X	88.07 (4.23)	88.04 (4.23)
	Var. inf. percentage of coverage for X	95.09 (16.36)	95.09 (16.36)
	Var. inf. comp. time (minutes)	0.85 (0.24)	0.84 (0.24)
	MCMC comp. time (minutes)	87.67 (0.29)	81.61 (0.39)
$\delta = 0.8$	Var. inf. vs MCMC accuracy for X	87.54 (5.17)	87.55 (5.20)
	Var. inf. percentage of coverage for X	94.01 (19.12)	94.01 (19.12)
	Var. inf. comp. time (minutes)	1.20 (0.28)	1.19 (0.28)
	MCMC comp. time (minutes)	88.08 (0.57)	82.12 (0.58)
$\delta = 0.9$	Var. inf. vs MCMC accuracy for X	87.10 (5.58)	87.12 (5.65)
	Var. inf. percentage of coverage for X	92.75 (22.02)	92.75 (22.02)
	Var. inf. comp. time (minutes)	1.43 (0.43)	1.43 (0.43)
	MCMC comp. time (minutes)	88.64 (0.30)	81.70 (0.44)

involving K in both cases with and without truncation. Once again, variational inference has been fully performed in R, while the main MCMC function has been implemented in R and uses C++ subroutines that replace the K and L matrix operations. Therefore, further computational advantages may be achieved implementing variational inference in C++.

6. Discussion

We have laid down the infrastructure for performing variational approximate inference on applications that can be studied through statistical inverse problems models. Our variational algorithms allow fast approximate fitting for these models, with satisfactory accuracy compared with standard MCMC.

The run-time of MCMC estimation for inverse problems is usually excessive on a standard personal computer. This means that parameter tuning, model diagnostics and sensitivity analyses are rarely performed. The use of variational inference methods, which are quick by comparison, means that multiple parameter settings and multiple models can be considered in a reasonable length of time. This opens-up the possibility of model diagnostics, such as influence and leverage, to become a routine part of applied inverse problems solution. Further, there is a subjective element, as always, to the choice of model components and in particular the hierarchical prior components. It would be a great advance for applications areas, such as medicine and archaeology, if rogue measurements could be identified and their influence on estimation could be quantified. Also, if any arbitrary parts could be shown to have insignificant effect on results, this would lead to far greater confidence and hence a wider acceptability of advanced statistical modelling approaches.

Hence, the implications of our work are not limited to the numerical results presented, but we provide a framework for other researchers to develop a richer set of model exploration methods for inverse problems. The flexibility of our approach is such that non-Normal likelihood distributions and other penalizations can be incorporated at will. Furthermore, this research sets the basis for several future directions to explore such as the study of settings with more than two dimensions, number of observations not matching that of data recording locations or more complex neighbour dependence structures.

Acknowledgements

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csda.2024.108055.

References

Agrawal, S., Kim, H., Sanz-Alonso, D., Strang, A., 2022. A variational inference approach to inverse problems with gamma hyperpriors. SIAM/ASA J. Uncertain. Quantificat. 10 (4), 1533–1559.

Allum, G., Aykroyd, R., Haigh, J., 1999. Empirical Bayes estimation for archaeological stratigraphy. J. R. Stat. Soc., Ser. C, Appl. Stat. 48 (1), 1–14.

Armagan, A., Dunson, D.B., Lee, J., 2013. Generalized double Pareto shrinkage. Stat. Sin. 23 (1), 119.

Arridge, S.R., Ito, K., Jin, B., Zhang, C., 2018. Variational Gaussian approximation for Poisson data. Inverse Probl. 34 (2), 025005.

Asif, A., Moura, J.M., 2005. Block matrices with l-block-banded inverse: inversion algorithms. IEEE Trans. Signal Process. 53 (2), 630–642.

Aykroyd, R.G., Haigh, J.G., Allum, G.T., 2001. Bayesian methods applied to survey data from archeological magnetometry. J. Am. Stat. Assoc. 96 (453), 64–76.

Bickel, P., Lindner, M., 2012. Approximating the inverse of banded matrices by banded matrices with applications to probability and statistics. Theory Probab. Appl. 56 (1), 1–20.

Carvalho, C.M., Polson, N.G., Scott, J.G., 2010. The horseshoe estimator for sparse signals. Biometrika 97 (2), 465–480.

Demko, S., Moss, W.F., Smith, P.W., 1984. Decay rates for inverses of band matrices. Math. Comput. 43 (168), 491–499.

Donoho, D.L., Johnstone, I.M., 1994. Ideal spatial adaptation by wavelet shrinkage. Biometrika 81 (3), 425–455.

Furrer, R., Sain, S.R., 2010. Spam: a sparse matrix R package with emphasis on mcmc methods for Gaussian Markov random fields. J. Stat. Softw. 36, 1–25.

Gehre, M., Jin, B., 2014. Expectation propagation for nonlinear inverse problems–with an application to electrical impedance tomography. J. Comput. Phys. 259, 513–535.

Georgiou, M., Fysikopoulos, E., Mikropoulos, K., Fragogeorgi, E., Loudos, G., 2017. Characterization of "γ-eye": a low-cost benchtop mouse-sized gamma camera for dynamic and static imaging studies. Mol. Imaging Biol. 19, 398–407.

Gerber, F., Mösinger, K., Furrer, R., 2017. Extending R packages to support 64-bit compiled code: an illustration with spam64 and gimms ndvi3g data. Comput. Geosci. 104, 109–119.

Green, P.J., 1990. Bayesian reconstructions from emission tomography data using a modified em algorithm. IEEE Trans. Med. Imaging 9 (1), 84–93.

Griffin, J.E., Brown, P.J., 2011. Bayesian hyper-lassos with non-convex penalization. Aust. N. Z. J. Stat. 53 (4), 423–442.

Guha, N., Wu, X., Efendiev, Y., Jin, B., Mallick, B.K., 2015. A variational Bayesian approach for inverse problems with skew-t error distributions. J. Comput. Phys. 301, 377–393.

Hadamard, J., 1902. Sur les problèmes aux dérivées partielles et leur signification physique. Princeton University Bulletin, pp. 49–52.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57, 97–109.

Kiebel, S.J., Daunizeau, J., Phillips, C., Friston, K.J., 2008. Variational Bayesian inversion of the equivalent current dipole model in eeg/meg. NeuroImage 39 (2), 728–741.

Kılıç, E., Stanica, P., 2013. The inverse of banded matrices. J. Comput. Appl. Math. 237 (1), 126–135.

Maestrini, L., Wand, M.P., 2018. Variational message passing for skew t regression. Stat 7 (1), e196.

Maestrini, L., Wand, M.P., 2021. The inverse G-Wishart distribution and variational message passing. Aust. N. Z. J. Stat. 63 (3), 517–541.

McGrory, C.A., Titterington, D.M., Reeves, R., Pettitt, A.N., 2009. Variational Bayes for estimating the parameters of a hidden Potts model. Stat. Comput. 19, 329–340.

McLean, M., Wand, M., 2019. Variational message passing for elaborate response regression models. Bayesian Anal. 14 (2), 371–398.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1092.

Nason, G., 2022. wavethresh: Wavelets Statistics and Transforms. R package version 4.7.2.

Nason, G.P., 2008. Wavelet Methods in Statistics with R. Springer.

Nathoo, F., Babul, A., Moiseev, A., Virji-Babul, N., Beg, M., 2014. A variational Bayes spatiotemporal model for electromagnetic brain mapping. Biometrics 70 (1), 132–143.

Neville, S.E., Ormerod, J.T., Wand, M., 2014. Mean field variational Bayes for continuous sparse signal shrinkage: pitfalls and remedies. Electron. J. Stat. 8 (1), 1113–1151.

Nolan, T.H., Wand, M.P., 2017. Accurate logistic variational message passing: algebraic and numerical details. Stat 6 (1), 102–112.

Ormerod, J.T., You, C., Müller, S., 2017. A variational Bayes approach to variable selection. Electron. J. Stat. 11 (2), 3549–3594.

Park, T., Casella, G., 2008. The Bayesian Lasso. J. Am. Stat. Assoc. 103 (482), 681–686.

Povala, J., Kazlauskaite, I., Febrianto, E., Cirak, F., Girolami, M., 2022. Variational Bayesian approximation of inverse problems using sparse precision matrices. Comput. Methods Appl. Mech. Eng. 393, 114712.

R Core Team, 2023. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Sato, M.-a., Yoshioka, T., Kajihara, S., Toyama, K., Goda, N., Doya, K., Kawato, M., 2004. Hierarchical Bayesian estimation for meg inverse problem. NeuroImage 23 (3), 806–826.

Tonolini, F., Radford, J., Turpin, A., Faccio, D., Murray-Smith, R., 2020. Variational inference for computational imaging inverse problems. J. Mach. Learn. Res. 21 (179), 1–46.

Tung, D.T., Tran, M.-N., Cuong, T.M., 2019. Bayesian adaptive lasso with variational Bayes for variable selection in high-dimensional generalized linear mixed models. Commun. Stat., Simul. Comput. 48 (2), 530–543.

Wand, M.P., 2017. Fast approximate inference for arbitrarily large semiparametric regression models via message passing. J. Am. Stat. Assoc. 112 (517), 137–168.

Wand, M.P., Ormerod, J.T., Padoan, S.A., Frührwirth, R., 2011. Mean field variational Bayes for elaborate distributions. Bayesian Anal. 6 (4), 847–900.

Weir, I.S., 1997. Fully Bayesian reconstructions from single-photon emission computed tomography data. J. Am. Stat. Assoc. 92 (437), 49–60.

Wijewardhana, U.L., Codreanu, M., 2016. A Bayesian approach for online recovery of streaming signals from compressive measurements. IEEE Trans. Signal Process. 65 (1), 184–199.

Wipf, D., Nagarajan, S., 2009. A unified Bayesian framework for MEG/EEG source imaging. NeuroImage 44 (3), 947–966.

Zhang, C., Arridge, S., Jin, B., 2019. Expectation propagation for Poisson data. Inverse Probl. 35 (8), 085006.

Further reading

Duong, T., 2024. ks: Kernel Smoothing. R package version 1.14.2.

Scollar, I., 1970. Fourier transform methods for the evaluation of magnetic maps. Prospez. Archeol. 5, 9–41.

Wickham, H., 2016. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag, New York.