Statistics and Computing
https://doi.org/10.1007/s11222-020-09924-y

Monte Carlo co-ordinate ascent variational inference

Lifeng Ye[1] · **Alexandros Beskos**[1] · **Maria De Iorio**[1,2] · **Jie Hao**[3]

Received: 10 May 2019 / Accepted: 22 January 2020
© The Author(s) 2020

Abstract
In variational inference (VI), coordinate-ascent and gradient-based approaches are two major types of algorithms for approximating difficult-to-compute probability densities. In real-world implementations of complex models, Monte Carlo methods are widely used to estimate expectations in coordinate-ascent approaches and gradients in derivative-driven ones. We discuss a Monte Carlo co-ordinate ascent VI (MC-CAVI) algorithm that makes use of Markov chain Monte Carlo (MCMC) methods in the calculation of expectations required within co-ordinate ascent VI (CAVI). We show that, under regularity conditions, an MC-CAVI recursion will get arbitrarily close to a maximiser of the evidence lower bound with any given high probability. In numerical examples, the performance of MC-CAVI algorithm is compared with that of MCMC and—as a representative of derivative-based VI methods—of Black Box VI (BBVI). We discuss and demonstrate MC-CAVI's suitability for models with *hard constraints* in simulated and real examples. We compare MC-CAVI's performance with that of MCMC in an important complex model used in nuclear magnetic resonance spectroscopy data analysis—BBVI is nearly impossible to be employed in this setting due to the hard constraints involved in the model.

Keywords Variational inference · Markov chain Monte Carlo · Coordinate-ascent · Gradient-based optimisation · Bayesian inference · Nuclear magnetic resonance

1 Introduction

Variational inference (VI) (Jordan et al. 1999; Wainwright et al. 2008) is a powerful method to approximate intractable integrals. As an alternative strategy to Markov chain Monte Carlo (MCMC) sampling, VI is fast, relatively straightforward for monitoring convergence and typically easier to scale to large data (Blei et al. 2017) than MCMC. The key idea of VI is to approximate difficult-to-compute conditional densities of latent variables, given observations, via use of optimization. A family of distributions is assumed for the latent variables, as an approximation to the exact conditional distribution. VI aims at finding the member, amongst the selected family, that minimizes the Kullback–Leibler (KL) divergence from the conditional law of interest.

Let x and z denote, respectively, the observed data and latent variables. The goal of the inference problem is to identify the conditional density (assuming a relevant reference measure, e.g. Lebesgue) of latent variables given observations, i.e. $p(z|x)$. Let $\mathcal{L}$ denote a family of densities defined over the space of latent variables—we denote members of this family as $q = q(z)$ below. The goal of VI is to find the element of the family closest in KL divergence to the true $p(z|x)$. Thus, the original inference problem can be rewritten as an optimization one: identify q^* such that

$$q^* = \underset{q\in\mathcal{L}}{\operatorname{argmin}}\, \mathrm{KL}(q \mid p(\cdot|x)) \tag{1}$$

for the KL-divergence defined as

$$\begin{aligned}\mathrm{KL}(q \mid p(\cdot|x)) &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z|x)] \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z,x)] + \log p(x),\end{aligned}$$

✉ Lifeng Ye
lifeng.ye.13@ucl.ac.uk

Alexandros Beskos
a.beskos@ucl.ac.uk

Maria De Iorio
maria@yale-nus.edu.sg

Jie Hao
j.hao@sjtu.edu.cn

[1] Department of Statistical Science, University College London, London, UK

[2] Yale-NUS College, Singapore, Singapore

[3] Key Laboratory of Systems Biomedicine (Ministry of Education), Shanghai Center for Systems Biomedicine, Shanghai Jiao Tong University, Shanghai, China

Published online: 14 February 2020

with $\log p(x)$ being constant w.r.t. z. Notation $\mathbb{E}_q$ refers to expectation taken over $z \sim q$. Thus, minimizing the KL divergence is equivalent to maximising the evidence lower bound, ELBO(q), given by

$$\text{ELBO}(q) = \mathbb{E}_q[\log p(z, x)] - \mathbb{E}_q[\log q(z)]. \tag{2}$$

Let $\mathsf{S}_p \subseteq \mathbb{R}^m, m \geq 1$, denote the support of the target $p(z|x)$, and $\mathsf{S}_q \subseteq \mathbb{R}^m$ the support of a variational density $q \in \mathcal{L}$—assumed to be common over all members $q \in \mathcal{L}$. Necessarily, $\mathsf{S}_p \subseteq \mathsf{S}_q$, otherwise the KL-divergence will diverge to $+\infty$.

Many VI algorithms focus on the mean-field variational family, where variational densities in $\mathcal{L}$ are assumed to factorise over blocks of z. That is,

$$\begin{aligned} q(z) = \prod_{i=1}^{b} q_i(z_i), \quad \mathsf{S}_q = \mathsf{S}_{q_1} \times \cdots \times \mathsf{S}_{q_b}, \\ z = (z_1, \ldots, z_b) \in \mathsf{S}_q, \quad z_i \in \mathsf{S}_{q_i}, \end{aligned} \tag{3}$$

for individual supports $\mathsf{S}_{q_i} \subseteq \mathbb{R}^{m_i}$, $m_i \geq 1$, $1 \leq i \leq b$, for some $b \geq 1$, and $\sum_i m_i = m$. It is advisable that highly correlated latent variables are placed in the same block to improve the performance of the VI method.

There are, in general, two types of approaches to maximise ELBO in VI: a co-ordinate ascent approach and a gradient-based one. Co-ordinate ascent VI (CAVI) (Bishop 2006) is amongst the most commonly used algorithms in this context. To obtain a local maximiser for ELBO, CAVI sequentially optimizes each factor of the mean-field variational density, while holding the others fixed. Analytical calculations on function space—involving variational derivatives—imply that, for given fixed $q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_b$, ELBO($q$) is maximised for

$$q_i(z_i) \propto \exp\left\{\mathbb{E}_{-i}[\log p(z_{i_-}, z_i, z_{i_+}, x)]\right\}, \tag{4}$$

where $z_{-i} := (z_{i_-}, z_{i_+})$ denotes vector z having removed component z_i, with i_- (resp. i_+) denoting the ordered indices that are smaller (resp. larger) than i; $\mathbb{E}_{-i}$ is the expectation taken under z_{-i} following its variational distribution, denoted q_{-i}. The above suggest immediately an iterative algorithm, guaranteed to provide values for ELBO(q) that cannot decrease as the updates are carried out.

The expected value $\mathbb{E}_{-i}[\log p(z_{i_-}, z_i, z_{i_+}, x)]$ can be difficult to derive analytically. Also, CAVI typically requires traversing the entire dataset at each iteration, which can be overly computationally expensive for large datasets. Gradient-based approaches, which can potentially scale up to large data—alluding here to recent Stochastic-Gradient-type methods—can be an effective alternative for ELBO optimisation. However, such algorithms have their own challenges, e.g. in the case of reparameterization Variational Bayes (VB) analytical derivation of gradients of the log-likelihood can often be problematic, while in the case of score-function VB the requirement of the gradient of $\log q$ restricts the range of the family $\mathcal{L}$ we can choose from.

In real-world applications, hybrid methods combining Monte Carlo with recursive algorithms are common, e.g., Auto-Encoding Variational Bayes, Doubly-Stochastic Variational Bayes for non-conjugate inference, Stochastic Expectation-Maximization (EM) (Beaumont et al. 2002; Sisson et al. 2007; Wei and Tanner 1990). In VI, Monte Carlo is often used to estimate the expectation within CAVI or the gradient within derivative-driven methods. This is the case, e.g., for Stochastic VI (Hoffman et al. 2013) and Black-Box VI (BBVI) (Ranganath et al. 2014).

BBVI is used in this work as a representative of gradient-based VI algorithms. It allows carrying out VI over a wide range of complex models. The variational density q is typically chosen within a parametric family, so finding q^* in (1) is equivalent to determining an optimal set of parameters that characterize $q_i = q_i(\cdot|\lambda_i)$, $\lambda_i \in \Lambda_i \subseteq \mathbb{R}^{d_i}$, $1 \leq d_i$, $1 \leq i \leq b$, with $\sum_{i=1}^{b} d_i = d$. The gradient of ELBO w.r.t. the variational parameters $\lambda = (\lambda_1, \ldots, \lambda_b)$ equals

$$\begin{aligned} \nabla_\lambda \text{ELBO}(q) := \mathbb{E}_q\big[\nabla_\lambda \log q(z|\lambda)\{\log p(z, x) \\ - \log q(z|\lambda)\}\big] \end{aligned} \tag{5}$$

and can be approximated by black-box Monte Carlo estimators as, e.g.,

$$\begin{aligned} \widehat{\nabla_\lambda \text{ELBO}}(q) := \frac{1}{N}\sum_{n=1}^{N}\big[\nabla_\lambda \log q(z^{(n)}|\lambda)\{\log p(z^{(n)}, x) \\ - \log q(z^{(n)}|\lambda)\}\big], \end{aligned} \tag{6}$$

with $z^{(n)} \overset{iid}{\sim} q(z|\lambda)$, $1 \leq n \leq N$, $N \geq 1$. The approximated gradient of ELBO can then be used within a stochastic optimization procedure to update λ at the kth iteration with

$$\lambda_{k+1} \leftarrow \lambda_k + \rho_k \widehat{\nabla_{\lambda_k} \text{ELBO}}(q), \tag{7}$$

where $\{\rho_k\}_{k\geq 0}$ is a Robbins-Monro-type step-size sequence (Robbins and Monro 1951). As we will see in later sections, BBVI is accompanied by generic variance reduction methods, as the variability of (6) for complex models can be large.

Remark 1 (Hard Constraints) Though gradient-based VI methods are some times more straightforward to apply than co-ordinate ascent ones,—e.g. combined with the use of modern approaches for automatic differentiation (Kucukelbir et al. 2017)—co-ordinate ascent methods can still be important for models with *hard constraints*, where gradient-based algorithms are laborious to apply. (We adopt the viewpoint here that one chooses variational densities that respect the

constraints of the target, for improved accuracy.) Indeed, notice in the brief description we have given above for CAVI and BBVI, the two methodologies are structurally different, as CAVI does not necessarily require to be be built via the introduction of an exogenous variational parameter λ. Thus, in the context of a support for the target $p(z|x)$ that involves complex constraints, a CAVI approach overcomes this issue naturally by blocking together the z_i's responsible for the constraints. In contrast, introduction of the variational parameter λ creates sometimes severe complications in the development of the derivative-driven algorithm, as normalising constants that depend on λ are extremely difficult to calculate analytically and obtain their derivatives. Thus, a main argument spanning this work—and illustrated within it—is that co-ordinate-ascent-based VI methods have a critical role to play amongst VI approaches for important classes of statistical models.

Remark 2 The discussion in Remark 1 is also relevant when VB is applied with constraints imposed on the *variational parameters*. E.g. the latter can involve covariance matrices, whence optimisation has to be carried out on the space of symmetric positive definite matrices. Recent attempts in the VB field to overcome this issue involves updates carried out on manifolds, see e.g. Tran et al. (2019).

The main contributions of the paper are:

(i) We discuss, and then apply a Monte Carlo CAVI (MC-CAVI) algorithm in a sequence of problems of increasing complexity, and study its performance. As the name suggests, MC-CAVI uses the Monte Carlo principle for the approximation of the difficult-to-compute conditional expectations, $\mathbb{E}_{-i}[\log p(z_{i_-}, z_i, z_{i_+}, x)]$, within CAVI.
(ii) We provide a justification for the algorithm by showing analytically that, under suitable regularity conditions, MC-CAVI will get arbitrarily close to a maximiser of the ELBO with high probability.
(iii) We contrast MC-CAVI with MCMC and BBVI through simulated and real examples, some of which involve hard constraints; we demonstrate MC-CAVI's effectiveness in an important application imposing such hard constraints, with real data in the context of Nuclear Magnetic Resonance (NMR) spectroscopy.

Remark 3 Inserting Monte Carlo steps within a VI approach (that might use a mean field or another approximation) is not uncommon in the VI literature. E.g., Forbes and Fort (2007) employ an MCMC procedure in the context of a Variational EM (VEM), to obtain estimates of the normalizing constant for Markov Random Fields—they provide asymptotic results for the correctness of the complete algorithm; Tran et al. (2016) apply Mean-Field Variational Bayes (VB) for Generalised Linear Mixed Models, and use Monte Carlo for the approximation of analytically intractable required expectations under the variational densities; several references for related works are given in the above papers. Our work focuses on MC-CAVI, and develops theory that is appropriate for this VI method. This algorithm has *not* been studied analytically in the literature, thus the development of its theoretical justification—even if it borrows elements from Monte Carlo EM—is new.

The rest of the paper is organised as follows. Section 2 presents briefly the MC-CAVI algorithm. It also provides—in a specified setting—an analytical result illustrating non-accumulation of Monte Carlo errors in the execution of the recursions of the algorithm. That is, with a probability arbitrarily close to 1, the variational solution provided by MC-CAVI can be as close as required to the one of CAVI, for a big enough Monte Carlo sample size, regardless of the number of algorithmic iterations. Section 3 shows two numerical examples, contrasting MC-CAVI with alternative algorithms. Section 4 presents an implementation of MC-CAVI in a real, complex, challenging posterior distribution arising in metabolomics. This is a practical application, involving hard constraints, chosen to illustrate the potential of MC-CAVI in this context. We finish with some conclusions in Sect. 5.

2 MC-CAVI algorithm

2.1 Description of the algorithm

We begin with a description of the basic CAVI algorithm. A double subscript will be used to identify block variational densities: $q_{i,k}(z_i)$ (resp. $q_{-i,k}(z_{-i})$) will refer to the density of the ith block (resp. all blocks but the ith), after k updates have been carried out on that block density (resp. k updates have been carried out on the blocks preceding the ith, and $k-1$ updates on the blocks following the ith).

- Step 0: Initialize probability density functions $q_{i,0}(z_i)$, $i = 1, \ldots, b$.
- Step k: For $k \geq 1$, given $q_{i,k-1}(z_i)$, $i = 1, \ldots, b$, execute:
  - For $i = 1, \ldots, b$, update:

    $$\log q_{i,k}(z_i) = const. + \mathbb{E}_{-i,k}[\log p(z, x)],$$

    with $\mathbb{E}_{-i,k}$ taken w.r.t. $z_{-i} \sim q_{-i,k}$.
- Iterate until convergence.

Assume that the expectations $\mathbb{E}_{-i}[\log p(z,x)]$, $\{i : i \in \mathcal{I}\}$, for an index set $\mathcal{I} \subseteq \{1,\ldots,b\}$, can be obtained analytically, over all updates of the variational density $q(z)$; and that this is not the case for $i \notin \mathcal{I}$. Intractable integrals can be approximated via a Monte Carlo method. (As we will see in the applications in the sequel, such a Monte Carlo device typically uses samples from an appropriate MCMC algorithm.) In particular, for $i \notin \mathcal{I}$, one obtains $N \geq 1$ samples from the current $q_{-i}(z_{-i})$ and uses the standard Monte Carlo estimate

$$\widehat{\mathbb{E}}_{-i}[\log p(z_{i_-}, z_i, z_{i_+}, x)] = \frac{\sum_{n=1}^{N} \log p(z_{i_-}^{(n)}, z_i, z_{i_+}^{(n)}, x)}{N}.$$

Implementation of such an approach gives rise to MC-CAVI, described in Algorithm 1.

Algorithm 1: MC-CAVI

Require: Number of iterations T.

Require: Number of Monte Carlo samples N.

Require: $\mathbb{E}_{-i}[\log p(z_{i_-}, z_i, z_{i_+}, x)]$ in closed form, for $i \in \mathcal{I}$.

Initialize $q_{i,0}(z_i)$, $i = 1,\ldots,b$.

for $k = 1 : T$ **do**

  for $i = 1 : b$ **do**

    If $i \in \mathcal{I}$, set $q_{i,k}(z_i) \propto \exp\left\{\mathbb{E}_{-i,k}[\log p(z_{i_-}, z_i, z_{i_+}, x)]\right\}$;

    If $i \notin \mathcal{I}$:

    Obtain N samples, $(z_{i_-,k}^{(n)}, z_{i_+,k-1}^{(n)})$, $1 \leq n \leq N$, from $q_{-i,k}(z_{-i})$.

    Set

$$q_{i,k}(z_i) \propto \exp\left\{\frac{\sum_{n=1}^{N} \log p(z_{i_-,k}^{(n)}, z_i, z_{i_+,k-1}^{(n)}, x)}{N}\right\}.$$

  end

end

2.2 Applicability of MC-CAVI

We discuss here the class of problems for which MC-CAVI can be applied. It is desirable to avoid settings where the order of samples or statistics to be stored in memory increases with the iterations of the algorithm. To set-up the ideas we begin with CAVI itself. Motivated by the standard exponential class of distributions, we work as follows.

Consider the case when the target density $p(z,x) \equiv f(z)$—we omit reference to the data x in what follows, as x is fixed and irrelevant for our purposes (notice that f is not required to integrate to 1)—is assumed to have the structure,

$$f(z) = h(z) \exp\left\{\langle \eta, T(z)\rangle - A(\eta)\right\}, \quad z \in \mathsf{S}_p, \tag{8}$$

for s-dimensional constant vector $\eta = (\eta_1,\ldots,\eta_s)$, vector function $T(z) = (T_1(z),\ldots,T_s(z))$, with some $s \geq 1$, and relevant scalar functions $h > 0$, A; $\langle\cdot,\cdot\rangle$ is the standard inner product in $\mathbb{R}^s$. Also, we are given the choice of block-variational densities $q_1(z_1),\ldots,q_b(z_b)$ in (3). Following the definition of CAVI from Sect. 2.1—assuming that the algorithm can be applied, i.e. all required expectations can be obtained analytically—the number of 'sufficient' statistics, say $T_{i,k}$ giving rise to the definition of $q_{i,k}$ will always be upper bounded by s. Thus, in our working scenario, CAVI will be applicable with a computational cost that is upper bounded by a constant within the class of target distributions in (8)—assuming relevant costs for calculating expectations remain bounded over the algorithmic iterations.

Moving on to MC-CAVI, following the definition of index set $\mathcal{I}$ in Sect. 2.1, recall that a Monte Carlo approach is required when updating $q_i(z_i)$ for $i \notin \mathcal{I}$, $1 \leq i \leq b$. In such a scenario, controlling computational costs amounts to having a target (8) admitting the factorisations,

$$h(z) \equiv h_i(z_i)h_{-i}(z_{-i}), \quad T_l(z) \equiv T_{l,i}(z_i)T_{l,-i}(z_{-i}),$$
$$1 \leq l \leq s, \quad \text{for all } i \notin \mathcal{I}. \tag{9}$$

Once (9) is satisfied, we do not need to store all N samples from $q_{-i}(z_{-i})$, but simply some relevant averages keeping the cost per iteration for the algorithm bounded. We stress that the combination of characterisations in (8)–(9) is very general and will typically be satisfied for most practical statistical models.

2.3 Theoretical justification of MC-CAVI

An advantageous feature of MC-CAVI versus derivative-driven VI methods is its structural similarity with Monte Carlo Expectation-Maximization (MCEM). Thus, one can build on results in the MCEM literature to prove asymptotical properties of MC-CAVI; see e.g. Chan and Ledolter (1995), Booth and Hobert (1999), Levine and Casella (2001), Fort and Moulines (2003). To avoid technicalities related with working on general spaces of probability density functions, we begin by assuming a parameterised setting for the variational densities—as in the BBVI case—with the family of variational densities being closed under CAVI or (more generally) MC-CAVI updates.

Assumption 1 (*Closedness of Parameterised $q(\cdot)$ Under Variational Update*) For the CAVI or the MC-CAVI algorithm, each $q_{i,k}(z_i)$ density obtained during the iterations of the algorithm, $1 \leq i \leq b$, $k \geq 0$, is of the parametric form

$$q_{i,k}(z_i) = q_i(z_i|\lambda_i^k),$$

for a unique $\lambda_i^k \in \Lambda_i \subseteq \mathbb{R}^{d_i}$, for some $d_i \geq 1$, for all $1 \leq i \leq b$. $\left(\text{Let } d = \sum_{i=1}^{b} d_i \text{ and } \Lambda = \Lambda_1 \times \cdots \times \Lambda_b.\right)$

Under Assumption 1, CAVI and MC-CAVI can be corresponded to some well-defined maps $M : \Lambda \mapsto \Lambda$, $\mathcal{M}_N : \Lambda \mapsto \Lambda$ respectively, so that, given current variational parameter λ, one step of the algorithms can be expressed in terms of a new parameter λ' (different for each case) obtained via the updates

$$\text{CAVI: } \lambda' = M(\lambda); \qquad \text{MC-CAVI: } \lambda' = \mathcal{M}_N(\lambda).$$

For an analytical study of the convergence properties of CAVI itself and relevant regularity conditions, see e.g. (Bertsekas 1999, Proposition 2.7.1), or numerous other resources in numerical optimisation. Expressing the MC-CAVI update—say, the $(k+1)$th one—as

$$\lambda^{k+1} = M(\lambda^k) + \{\mathcal{M}_N(\lambda^k) - M(\lambda^k)\}, \tag{10}$$

it can be seen as a random perturbation of a CAVI step. In the rest of this section we will explore the asymptotic properties of MC-CAVI. We follow closely the approach in Chan and Ledolter (1995)—as it provides a less technical procedure, compared e.g. to Fort and Moulines (2003) or other works about MCEM—making all appropriate adjustments to fit the derivations into the setting of the MC-CAVI methodology along the way. We denote by M^k, $\mathcal{M}_N^k$, the k-fold composition of M, $\mathcal{M}_N$ respectively, for $k \geq 0$.

Assumption 2 Λ is an open subset of $\mathbb{R}^d$, and the mappings $\lambda \mapsto \text{ELBO}(q(\lambda))$, $\lambda \mapsto M(\lambda)$ are continuous on Λ.

If $M(\lambda) = \lambda$ for some $\lambda \in \Lambda$, then λ is a fixed point of $M()$. A given $\lambda^* \in \Lambda$ is called an isolated local maximiser of the $\text{ELBO}(q(\cdot))$ if there is a neighborhood of λ^* over which λ^* is the unique maximiser of the $\text{ELBO}(q(\cdot))$.

Assumption 3 (*Properties of* $M(\cdot)$ *Near a Local Maximum*) *Let* $\lambda^* \in \Lambda$ *be an isolated local maximum of ELBO*$(q(\cdot))$. *Then,*

(i) λ^* *is a fixed point of* $M(\cdot)$;
(ii) *there is a neighborhood* $V \subseteq \Lambda$ *of* λ^* *over which* λ^* *is a unique maximum, such that* $\text{ELBO}(q(M(\lambda))) > \text{ELBO}(q(\lambda))$ *for any* $\lambda \in V \backslash \{\lambda^*\}$.

Notice that the above assumption refers to the deterministic update $M(\cdot)$, which performs co-ordinate ascent; thus requirements (i), (ii) are fairly weak for such a recursion. The critical technical assumption required for delivering the convergence results in the rest of this section is the following one.

Assumption 4 (*Uniform Convergence in Probability on Compact Sets*) For any compact set $C \subseteq \Lambda$ the following holds: for any $\varrho, \varrho' > 0$, there exists a positive integer N_0, such that for all $N \geq N_0$ we have,

$$\inf_{\lambda \in C} \text{Prob}\left[\, |\mathcal{M}_N(\lambda) - M(\lambda)| < \varrho \,\right] > 1 - \varrho'.$$

It is beyond the context of this paper to examine Assumption 4 in more depth. We will only stress that Assumption 4 is the sufficient structural condition that allows to extend closeness between CAVI and MC-CAVI updates in a single algorithmic step into one for arbitrary number of steps.

We continue with a definition.

Definition 1 A fixed point λ^* of $M(\cdot)$ is said to be asymptotically stable if,

(i) for any neighborhood V_1 of λ^*, there is a neighborhood V_2 of λ^* such that for all $k \geq 0$ and all $\lambda \in V_2$, $M^k(\lambda) \in V_1$;
(ii) there exists a neighbourhood V of λ^* such that $\lim_{k\to\infty} M^k(\lambda) = \lambda^*$ if $\lambda \in V$.

We will state the main asymptotic result for MC-CAVI in Theorem 1 that follows; first we require Lemma 1.

Lemma 1 *Let Assumptions 1–3 hold. If* λ^* *is an isolated local maximiser of ELBO*$(q(\cdot))$, *then* λ^* *is an asymptotically stable fixed point of* $M(\cdot)$.

The main result of this section is as follows.

Theorem 1 *Let Assumptions 1–4 hold and* λ^* *be an isolated local maximiser of* $\text{ELBO}(q(\cdot))$. *Then there exists a neighbourhood, say* V_1, *of* λ^* *such that for starting values* $\lambda \in V_1$ *of MC-CAVI algorithm and for all* $\epsilon_1 > 0$, *there exists a* k_0 *such that*

$$\lim_{N\to\infty} \text{Prob}\left(|\mathcal{M}_N^k - \lambda^*| < \epsilon_1 \text{ for some } k \leq k_0\right) = 1.$$

The proofs of Lemma 1 and Theorem 1 can be found in "Appendices A and B", respectively.

2.4 Stopping criterion and sample size

The method requires the specification of the Monte Carlo size N and a stopping rule.

Principled: but impractical—approach

As the algorithm approaches a local maximum, changes in ELBO should be getting closer to zero. To evaluate the performance of MC-CAVI, one could, in principle, attempt to monitor the evolution of ELBO during the algorithmic iterations. For current variational distribution $q = (q_1, \ldots, q_b)$, assume that MC-CAVI is about to update q_i with $q_i' = q_{i,N}'$, where the addition of the second subscript at this point emphasizes the dependence of the new value for q_i on the Monte Carlo size N. Define,

$$\Delta\text{ELBO}(q, N) = \text{ELBO}(q_{i-}, q_{i,N}', q_{i+}) - \text{ELBO}(q).$$

If the algorithm is close to a local maximum, $\Delta\text{ELBO}(q, N)$ should be close to zero, at least for sufficiently large N. Given such a choice of N, an MC-CAVI recursion should be terminated once $\Delta\text{ELBO}(q, N)$ is smaller than a user-specified tolerance threshold. Assume that the random variable $\Delta\text{ELBO}(q, N)$ has mean $\mu = \mu(q, N)$ and variance $\sigma^2 = \sigma^2(q, N)$. Chebychev's inequality implies that, with probability greater than or equal to $(1 - 1/K^2)$, $\Delta\text{ELBO}(q, N)$ lies within the interval $(\mu - K\sigma, \mu + K\sigma)$, for any real $K > 0$. Assume that one fixes a large enough K. The choice of N and of a stopping criterion should be based on the requirements:

(i) $\sigma \leq \nu$, with ν a predetermined level of tolerance;
(ii) the effective range $(\mu - K\sigma, \mu + K\sigma)$ should include zero, implying that $\Delta\text{ELBO}(q, N)$ differs from zero by less than $2K\sigma$.

Requirement (i) provides a rule for the choice of N—assuming applied over all $1 \leq i \leq b$, for q in areas close to a maximiser,—and requirement (ii) a rule for defining a stopping criterion. Unfortunately, the above considerations—based on the proper term $\text{ELBO}(q)$ that VI aims to maximise—involve quantities that are typically impossible to obtain analytically or via some reasonably expensive approximation.

Practical considerations

Similarly to MCEM, it is recommended that N gets increased as the algorithm becomes more stable. It is computationally inefficient to start with a large value of N when the current variational distribution can be far from the maximiser. In practice, one may monitor the convergence of the algorithm by plotting relevant *statistics* of the variational distribution versus the number of iterations. We can declare that convergence has been reached when such traceplots show relatively small random fluctuations (due to the Monte Carlo variability) around a fixed value. At this point, one may terminate the algorithm or continue with a larger value of N, which will further decrease the traceplot variability. In the applications we encounter in the sequel, we typically have $N \leq 100$, so calculating, for instance, Effective Sample Sizes to monitor the mixing performance of the MCMC steps is not practical.

3 Numerical examples: simulation study

In this section we illustrate MC-CAVI with two simulated examples. First, we apply MC-CAVI and CAVI on a simple model to highlight main features and implementation strategies. Then, we contrast MC-CAVI, MCMC, BBVI in a complex scenario with hard constraints.

3.1 Simulated example 1

We generate $n = 10^3$ data points from $\text{N}(10, 100)$ and fit the semi-conjugate Bayesian model

Example Model 1

$$\begin{aligned} x_1, \ldots, x_n &\sim \text{N}(\vartheta, \tau^{-1}), \\ \vartheta &\sim \text{N}(0, \tau^{-1}), \\ \tau &\sim \text{Gamma}(1, 1). \end{aligned}$$

Let $\bar{x}$ be the data sample mean. In each iteration, the CAVI density function—see (4)—for τ is that of the Gamma distribution $\text{Gamma}(\frac{n+3}{2}, \zeta)$, with

$$\zeta = 1 + \frac{(1+n)\mathbb{E}(\vartheta^2) - 2(n\bar{x})\mathbb{E}(\vartheta) + \sum_{j=1}^n x_j^2}{2},$$

whereas for ϑ that of the normal distribution $\text{N}(\frac{n\bar{x}}{1+n}, \frac{1}{(1+n)\mathbb{E}(\tau)})$.

$(\mathbb{E}(\vartheta), \mathbb{E}(\vartheta^2))$ and $\mathbb{E}(\tau)$ denote the relevant expectations under the current CAVI distributions for ϑ and τ respectively; the former are initialized at 0—there is no need to initialise $\mathbb{E}(\tau)$ in this case. Convergence of CAVI can be monitored, e.g., via the sequence of values of $\theta := (1 + n)\mathbb{E}(\tau)$ and ζ. If the change in values of these two parameters is smaller than, say, 0.01%, we declare convergence. Figure 1 shows the traceplots of θ, ζ.

Convergence is reached within 0.0017 s,[1] after precisely two iterations, due to the simplicity of the model. The resulted CAVI distribution for ϑ is $\text{N}(9.6, 0.1)$, and for τ it is $\text{Gamma}(501.5, 50130.3)$ so that $\mathbb{E}(\tau) \approx 0.01$.

Assume now that $q(\tau)$ was intractable. Since $\mathbb{E}(\tau)$ is required to update the approximate distribution of ϑ, an MCMC step can be employed to sample $\tau_1, \ldots, \tau_N$

[1] A Dell Latitude E5470 with Intel(R) Core(TM) i5-6300U CPU@2.40GHz is used for all experiments in this paper.

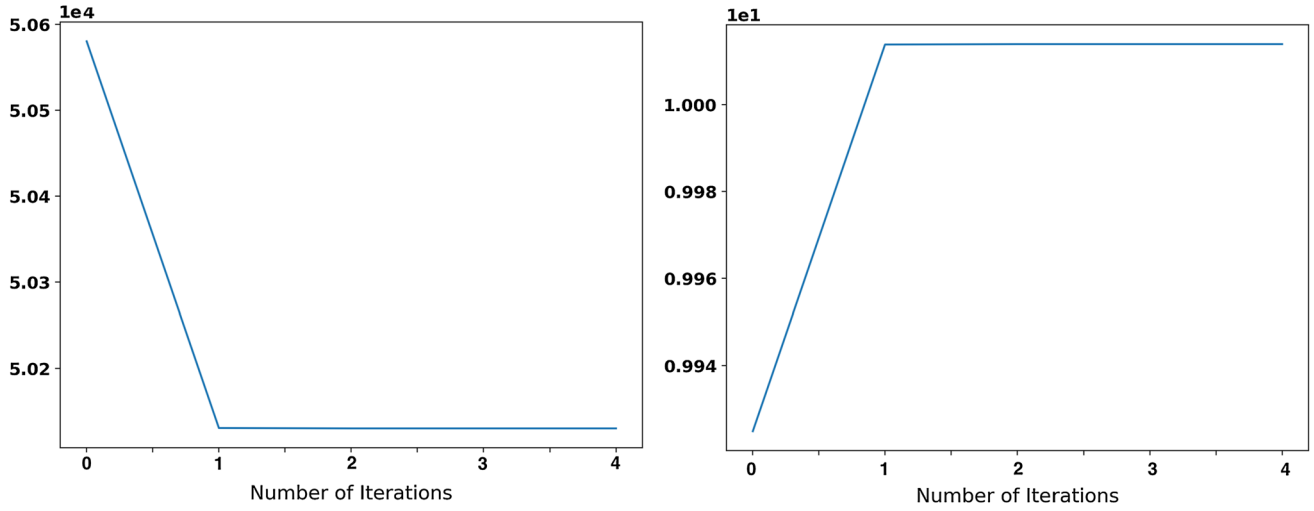

Fig. 1 Tracplots of ζ (left), θ (right) from application of CAVI on Simulated Example 1

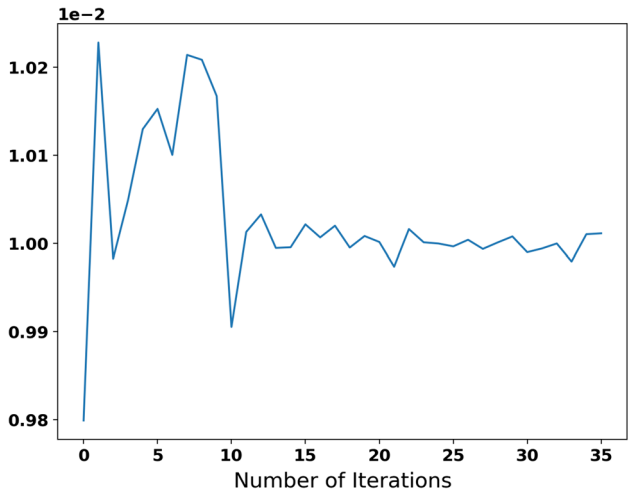

Fig. 2 Traceplot of $\widehat{\mathbb{E}}(\tau)$ generated by MC-CAVI for Simulated Example 1, using $N = 10$ for the first 10 iterations of the algorithm, and $N = 10^3$ for the rest. The y-axis gives the values of $\widehat{\mathbb{E}}(\tau)$ across iterations

from $q(\tau)$ to produce the Monte Carlo estimator $\widehat{\mathbb{E}}(\tau) = \sum_{j=1}^{N} \tau_j / N$. Within this MC-CAVI setting, $\widehat{\mathbb{E}}(\tau)$ will replace the exact $\mathbb{E}(\tau)$ during the algorithmic iterations. $(\mathbb{E}(\vartheta), \mathbb{E}(\vartheta^2))$ are initialised as in CAVI. For the first 10 iterations we set $N = 10$, and for the remaining ones, $N = 10^3$ to reduce variability. We monitor the values of $\widehat{\mathbb{E}}(\tau)$ shown in Fig. 2. The figure shows that MC-CAVI has stabilized after about 15 iterations; algorithmic time was 0.0114 s. To remove some Monte Carlo variability, the final estimator of $\mathbb{E}(\tau)$ is produced by averaging the last 10 values of its traceplot, which gives $\widehat{\mathbb{E}}(\tau) = 0.01$, i.e. a value very close to the one obtained by CAVI. The estimated distribution of ϑ is N(9.6, 0.1), the same as with CAVI.

The performance of MC-CAVI depends critically on the choice N. Let A be the value of N in the burn-in period, B the number of burn-in iterations and C the value of N after burn-in. Figure 3 shows trace plots of $\widehat{\mathbb{E}}(\tau)$ under different settings of the triplet A–B–C.

As with MCEM, N should typically be set to a small number at the beginning of the iterations so that the algorithm can reach fast a region of relatively high probability. N should then be increased to reduce algorithmic variability close to the convergence region. Figure 4 shows plots of convergence time versus variance of $\widehat{\mathbb{E}}(\tau)$ (left panel) and versus N (right panel). In VI, iterations are typically terminated when the (absolute) change in the monitored estimate is less than a small threshold. In MC-CAVI the estimate fluctuates around the limiting value after convergence (Table 1). In the simulation in Fig. 4, we terminate the iterations when the difference between the estimated mean (disregarding the first half of the chain) and the true value (0.01) is less than 10^{-5}. Figure 4 shows that: (i) convergence time decreases when the variance of $\widehat{\mathbb{E}}(\tau)$ decreases, as anticipated; (ii) convergence time decreases when N increases. In (ii), the decrease is most evident when N is still relatively small After N exceeds 200, convergence time remains almost fixed, as the benefit brought by decrease of variance is offset by the cost of extra samples. (This is also in agreement with the policy of N set to a small value at the initial iterations of the algorithm.)

3.2 Variance reduction for BBVI

In non-trivial applications, the variability of the initial estimator $\nabla_\lambda \widehat{\mathrm{ELBO}}(q)$ within BBVI in (6) will typically be large, so variance reduction approaches such as Rao-Blackwellization and control variates (Ranganath et al. 2014) are also used. Rao-Blackwellization (Casella and Robert 1996) reduces

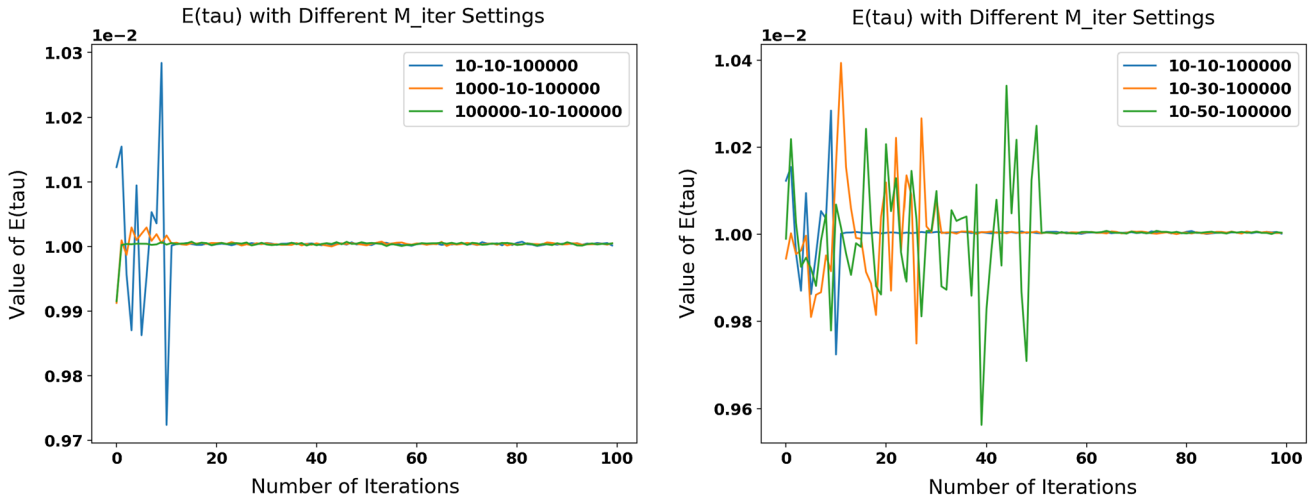

Fig. 3 Traceplot of $\widehat{\mathbb{E}}(\tau)$ under different settings of A–B–C (respectively, the value of N in the burn-in period, the number of burn-in iterations and the value of N after burn-in) for Simulated Example 1

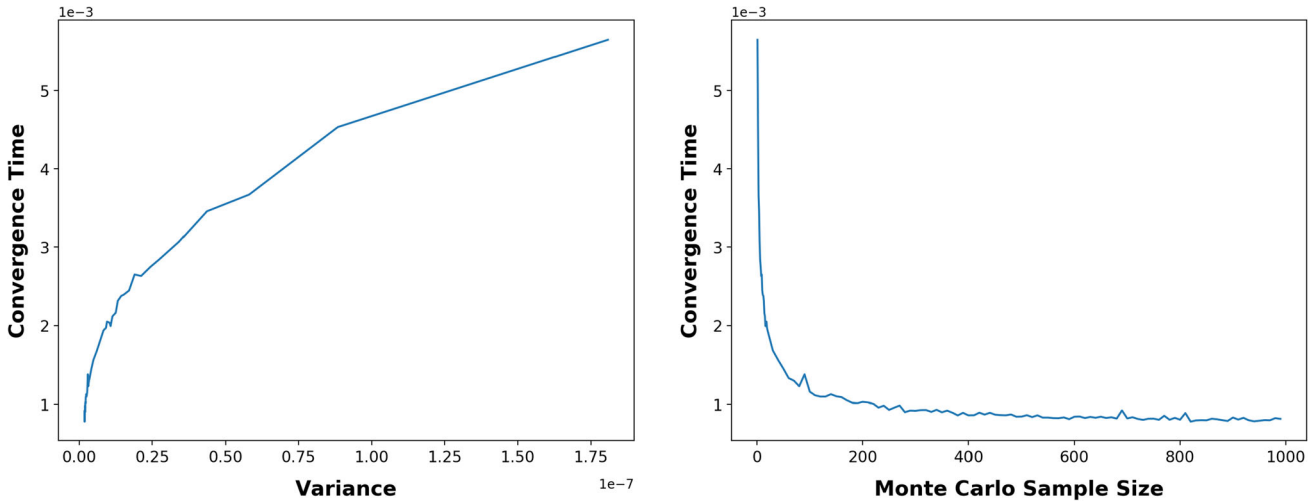

Fig. 4 Plot of convergence time versus variance of $\widehat{\mathbb{E}}(\tau)$ (left panel) and versus Monte Carlo sample size N (right panel)

Table 1 Results of MC-CAVI for Simulated Example 1

A–B–C	10–10–10^5	10^3–10–10^5	10^5–10–10^5	10–30–10^5	10–50–10^5
Time (s)	0.4640	0.4772	0.5152	0.3573	0.2722
$\widehat{\mathbb{E}}(\tau)$	0.01	0.01	0.01	0.01	0.01

variances by analytically calculating conditional expectations. In BBVI, within the factorization framework of (3), where $\lambda = (\lambda_1, \ldots, \lambda_b)$, and recalling identity (5) for the gradient, a Monte Carlo estimator for the gradient with respect to λ_i, $i \in \{1, \ldots, b\}$, can be simplified as

$$\nabla_{\lambda_i} \widehat{\mathrm{ELBO}}(q_i) = \frac{1}{N} \sum_{n=1}^{N} \big[\nabla_{\lambda_i} \log q_i(z_i^{(n)}|\lambda_i)\{\log c_i(z_i^{(n)}, x) - \log q_i(z_i^{(n)}|\lambda_i)\}\big], \tag{11}$$

with $z_i^{(n)} \overset{iid}{\sim} q_i(z_i|\lambda_i)$, $1 \le n \le N$, and,

$$c_i(z_i, x) := \exp\left\{\mathbb{E}_{-i}[\log p(z_{i_-}, z_i, z_{i_+}, x)]\right\}.$$

Depending on the model at hand, term $c_i(z_i, x)$ can be obtained analytically or via a double Monte Carlo procedure (for estimating $c_i(z_i^{(n)}, x)$, over all $1 \le n \le N$)—or a combination of thereof. In BBVI, control variates (Ross 2002) can be defined on a per-component basis and be applied to the Rao-Blackwellized noisy gradients of ELBO in (11) to

provide the estimator,

$$\nabla_{\lambda_i}\widehat{\mathrm{ELBO}}(q_i) = \frac{1}{N}\sum_{n=1}^{N}\big[\nabla_{\lambda_i}\log q_i(z_i^{(n)}|\lambda_i)\{\log c_i(z_i^{(n)},x) - \log q_i(z_i^{(n)}|\lambda_i) - \widehat{a}_i^*\}\big], \tag{12}$$

for the control,

$$\widehat{a}_i^* := \frac{\sum_{j=1}^{d_i}\widehat{\mathrm{Cov}}(f_{i,j},g_{i,j})}{\sum_{j=1}^{d_i}\widehat{\mathrm{Var}}(g_{i,j})},$$

where $f_{i,j}$, $g_{i,j}$ denote the jth co-ordinate of the vector-valued functions f_i, g_i respectively, given below,

$$g_i(z_i) := \nabla_{\lambda_i}\log q_i(z_i|\lambda_i),$$
$$f_i(z_i) := \nabla_{\lambda_i}\log q_i(z_i|\lambda_i)\{\log c_i(z_i,x) - \log q_i(z_i|\lambda_i)\}.$$

3.3 Simulated example 2: model with hard constraints

In this section, we discuss the performance and challenges of MC-CAVI, MCMC, BBVI for models where the support of the posterior—thus, also the variational distribution—involves hard constraints.

Here, we provide an example which offers a simplified version of the NMR problem discussed in Sect. 4 but allows for the implementation of BBVI, as the involved normalising constants can be easily computed. Moreover, as with other gradient-based methods, BBVI requires to tune the step-size sequence $\{\rho_k\}$ in (7), which might be a laborious task, in particular for increasing dimension. Although there are several proposals aimed to optimise the choice of $\{\rho_k\}$ (Bottou 2012; Kucukelbir et al. 2017), MC-CAVI does not face such a tuning requirement.

We simulate data according to the following scheme: observations $\{y_j\}$ are generated from $\mathrm{N}(\vartheta+\kappa_j,\theta^{-1})$, $j=1,\ldots,n$, with $\vartheta=6$, $\kappa_j = 1.5\cdot\sin(-2\pi+4\pi(j-1)/n)$, $\theta=3$, $n=100$. We fit the following model:

Example Model 2

$$\begin{aligned} y_j \mid \vartheta,\kappa_j,\theta &\sim \mathrm{N}(\vartheta+\kappa_j,\theta^{-1}),\\ \vartheta &\sim \mathrm{N}(0,10),\\ \kappa_j \mid \psi_j &\sim \mathrm{TN}(0,10,-\psi_j,\psi_j),\\ \psi_j &\overset{i.i.d.}{\sim} \mathrm{TN}(0.05,10,0,2),\quad j=1,\ldots,n,\\ \theta &\sim \mathrm{Gamma}(1,1). \end{aligned}$$

MCMC

We use a standard Metropolis-within-Gibbs. We set $y=(y_1,\ldots,y_n)$, $\kappa=(\kappa_1,\ldots,\kappa_n)$ and $\psi=(\psi_1,\ldots,\psi_n)$. Notice that we have the full conditional distributions,

$$p(\vartheta|y,\theta,\kappa,\psi) = \mathrm{N}\left(\frac{\sum_{j=1}^n(y_j-\kappa_j)\theta}{\frac{1}{10}+n\theta},\frac{1}{\frac{1}{10}+n\theta}\right),$$
$$p(\kappa_j|y,\theta,\vartheta,\psi) = \mathrm{TN}\left(\frac{(y_j-\vartheta)\theta}{\frac{1}{10}+\theta},\frac{1}{\frac{1}{10}+\theta},-\psi_j,\psi_j\right),$$
$$p(\theta|y,\vartheta,\kappa,\psi) = \mathrm{Gamma}\left(1+\frac{n}{2},1+\frac{\sum_{j=1}^n(y_j-\vartheta-\kappa_j)^2}{2}\right).$$

(Above, and in similar expressions written in the sequel, equality is meant to be properly understood as stating that 'the density on the left is equal to the density of the distribution on the right'.) For each ψ_j, $1\le j\le n$, the full conditional is,

$$p(\psi_j|y,\theta,\vartheta,\kappa) \propto \frac{\phi(\frac{\psi_j-\frac{1}{20}}{\sqrt{10}})}{\Phi(\frac{\psi_j}{\sqrt{10}})-\Phi(\frac{-\psi_j}{\sqrt{10}})}\mathbb{I}[\,|\kappa_j|<\psi_j<2\,],\quad j=1,\ldots,n,$$

where $\phi(\cdot)$ is the density of $\mathrm{N}(0,1)$ and $\Phi(\cdot)$ its cdf. The Metropolis–Hastings proposal for ψ_j is a uniform variate from $\mathrm{U}(0,2)$.

MC-CAVI

For MC-CAVI, the logarithm of the joint distribution is given by,

$$\begin{aligned}\log p(y,\vartheta,\kappa,\psi,\theta) = const. &+ \frac{n}{2}\log\theta - \frac{\theta\sum_{j=1}^n(y_j-\vartheta-\kappa_j)^2}{2}\\ &- \frac{\vartheta^2}{2\cdot 10} - \theta - \sum_{j=1}^n\frac{\kappa_j^2+(\psi_j-\frac{1}{20})^2}{2\cdot 10}\\ &- \sum_{j=1}^n\log(\Phi(\frac{\psi_j}{\sqrt{10}})-\Phi(\frac{-\psi_j}{\sqrt{10}})),\end{aligned}$$

under the constraints,

$$|\kappa_j|<\psi_j<2,\quad j=1,\ldots,n.$$

To comply with the above constraints, we factorise the variational distribution as,

$$q(\vartheta,\theta,\kappa,\psi) = q(\vartheta)q(\theta)\prod_{j=1}^n q(\kappa_j,\psi_j). \tag{13}$$

Here, for the relevant iteration k, we have,

$$q_k(\vartheta) = \mathrm{N}\left(\frac{\sum_{j=1}^n(y_j-\mathbb{E}_{k-1}(\kappa_j))\mathbb{E}_{k-1}(\theta)}{\frac{1}{10}+n\mathbb{E}_{k-1}(\theta)},\frac{1}{\frac{1}{10}+n\mathbb{E}_{k-1}(\theta)}\right),$$
$$q_k(\theta) = \mathrm{Gamma}\big(1+\frac{n}{2},1+\frac{\sum_{j=1}^n\mathbb{E}_{k,k-1}((y_j-\vartheta-\kappa_j)^2)}{2}\big)\big),$$
$$q_k(\kappa_j,\psi_j) \propto \exp\big\{-\frac{\mathbb{E}_k(\theta)(\kappa_j-(y_j-\mathbb{E}_k(\vartheta)))^2}{2}$$

$$- \frac{\kappa_j^2 + (\psi_j - \frac{1}{20})^2}{2 \cdot 10} \Big\} \Big/ \Big(\Phi\big(\frac{\psi_j}{\sqrt{10}}\big) - \Phi\big(\frac{-\psi_j}{\sqrt{10}}\big) \Big)$$
$$\cdot \mathbb{I}[\,|\kappa_j| < \psi_j < 2\,], \qquad 1 \le j \le n.$$

The quantity $\mathbb{E}_{k,k-1}((y_j - \vartheta - \kappa_j)^2)$ used in the second line above means that the expectation is considered under $\vartheta \sim q_k(\vartheta)$ and (independently) $\kappa_j \sim q_{k-1}(\kappa_j, \psi_j)$.

Then, MC-CAVI develops as follows:

- Step 0: For $k = 0$, initialize $\mathbb{E}_0(\theta) = 1$, $\mathbb{E}_0(\vartheta) = 4$, $\mathbb{E}_0(\vartheta^2) = 17$.
- Step k: For $k \ge 1$, given $\mathbb{E}_{k-1}(\theta)$, $\mathbb{E}_{k-1}(\vartheta)$, execute:
  - For $j = 1, \ldots, n$, apply an MCMC algorithm—with invariant law $q_{k-1}(\kappa_j, \psi_j)$—consisted of a number, N, of Metropolis-within-Gibbs iterations carried out over the relevant full conditionals,

$$q_{k-1}(\psi_j|\kappa_j) \propto \frac{\phi(\frac{\psi_j - \frac{1}{20}}{\sqrt{10}})}{\Phi(\frac{\psi_j}{\sqrt{10}}) - \Phi(\frac{-\psi_j}{\sqrt{10}})} \mathbb{I}[\,|\kappa_j| < \psi_j < 2\,],$$
$$q_{k-1}(\kappa_j|\psi_j) = \mathrm{TN}\big(\tfrac{(y_j - \mathbb{E}_{k-1}(\vartheta))\mathbb{E}_{k-1}(\theta)}{\frac{1}{10} + \mathbb{E}_{k-1}(\theta)}, \tfrac{1}{\frac{1}{10} + \mathbb{E}_{k-1}(\theta)}, - \psi_j, \psi_j\big).$$

    As with the full conditional $p(\psi_j|y, \theta, \vartheta, \kappa)$ within the MCMC sampler, we use a uniform proposal U(0, 2) at the Metropolis–Hastings step applied for $q_{k-1}(\psi_j|\kappa_j)$. For each k, the N iterations begin from the (κ_j, ψ_j)-values obtained at the end of the corresponding MCMC iterations at step $k-1$, with very first initial values being $\kappa, \psi_j) = (0, 1)$. Use the N samples to obtain $\mathbb{E}_{k-1}(\kappa_j)$ and $\mathbb{E}_{k-1}(\kappa_j^2)$.
  - Update the variational distribution for ϑ,

$$q_k(\vartheta) = \mathrm{N}\left(\frac{\sum_{j=i}^n (y_j - \mathbb{E}_{k-1}(\kappa_j))\mathbb{E}_{k-1}(\theta)}{\frac{1}{10} + n\mathbb{E}_{k-1}(\theta)}, \frac{1}{\frac{1}{10} + n\mathbb{E}_{k-1}(\theta)} \right)$$

    and evaluate $\mathbb{E}_k(\vartheta)$, $\mathbb{E}_k(\vartheta^2)$.
  - Update the variational distribution for θ,

$$q_k(\theta) = \mathrm{Gamma}\big(1 + \tfrac{n}{2}, 1 + \tfrac{\sum_{j=1}^n \mathbb{E}_{k,k-1}((y_j - \vartheta - \kappa_j)^2)}{2}\big)$$

    and evaluate $\mathbb{E}_k(\theta)$.
- Iterate until convergence.

BBVI

For BBVI we assume a variational distribution $q(\theta, \vartheta, \kappa, \psi \,|\, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ that factorises as in the case of CAVI in (13), where

$$\boldsymbol{\alpha} = (\alpha_\vartheta, \alpha_\theta, \alpha_{\kappa_1}, \ldots, \alpha_{\kappa_n}, \alpha_{\psi_1}, \ldots, \alpha_{\psi_n}),$$
$$\boldsymbol{\gamma} = (\gamma_\vartheta, \gamma_\theta, \gamma_{\kappa_1}, \ldots, \gamma_{\kappa_n}, \gamma_{\psi_1}, \ldots, \gamma_{\psi_n})$$

to be the variational parameters. Individual marginal distributions are chosen to agree—in type—with the model priors. In particular, we set,

$$q(\vartheta) = \mathrm{N}(\alpha_\vartheta, \exp(\gamma_\vartheta)),$$
$$q(\theta) = \mathrm{Gamma}(\exp(\alpha_\theta), \exp(\gamma_\theta)),$$
$$q(\kappa_j, \psi_j) = \mathrm{TN}(\alpha_{\kappa_j}, \exp(2\gamma_{\kappa_j}), - \psi_j, \psi_j) \otimes \mathrm{TN}(\alpha_{\psi_j}, \exp(2\gamma_{\psi_j}), 0, 2), \quad 1 \le j \le n.$$

It is straightforward to derive the required gradients (see "Appendix C" for the analytical expressions). BBVI is applied using Rao-Blackwellization and control variates for variance reduction. The algorithm is as follows,

- Step 0: Set $\eta = 0.5$; initialise $\boldsymbol{\alpha}^0 = 0$, $\boldsymbol{\gamma}^0 = 0$ with the exception $\alpha_\vartheta^0 = 4$.
- Step k: For $k \ge 1$, given $\boldsymbol{\alpha}^{k-1}$ and $\boldsymbol{\gamma}^{k-1}$ execute:
  - Draw $(\vartheta^i, \theta^i, \kappa^i, \psi^i)$, for $1 \le i \le N$, from $q_{k-1}(\vartheta)$, $q_{k-1}(\theta)$, $q_{k-1}(\kappa, \psi)$.
  - With the samples, use (12) to evaluate:

$$\nabla^k_{\alpha_\vartheta} \widehat{\mathrm{ELBO}}(q(\vartheta)), \quad \nabla^k_{\gamma_\vartheta} \widehat{\mathrm{ELBO}}(q(\vartheta)),$$
$$\nabla^k_{\alpha_\theta} \widehat{\mathrm{ELBO}}(q(\theta)), \quad \nabla^k_{\gamma_\theta} \widehat{\mathrm{ELBO}}(q(\theta)),$$
$$\nabla^k_{\alpha_{\kappa_j}} \widehat{\mathrm{ELBO}}(q(\kappa_j, \psi_j)), \quad \nabla^k_{\gamma_{\kappa_j}} \widehat{\mathrm{ELBO}}(q(\kappa_j, \psi_j)), \quad 1 \le j \le n,$$
$$\nabla^k_{\alpha_{\psi_j}} \widehat{\mathrm{ELBO}}(q(\kappa_j, \psi_j)), \quad \nabla^k_{\gamma_{\psi_j}} \widehat{\mathrm{ELBO}}(q(\kappa_j, \psi_j)), \quad 1 \le j \le n.$$

    (Here, superscript k at the gradient symbol ∇ specifies the BBVI iteration.)
  - Evaluate $\boldsymbol{\alpha}^k$ and $\boldsymbol{\gamma}^k$:

$$(\boldsymbol{\alpha}, \boldsymbol{\gamma})^k = (\boldsymbol{\alpha}, \boldsymbol{\gamma})^{k-1} + \rho_k \nabla^k_{(\boldsymbol{\alpha}, \boldsymbol{\gamma})} \widehat{\mathrm{ELBO}}(q),$$

    where $q = (q(\vartheta), q(\theta), q(\kappa_1, \psi_1), \ldots, q(\kappa_n, \psi_n))$. For the learning rate, we employed the AdaGrad algorithm (Duchi et al. 2011) and set $\rho_k = \eta\, \mathrm{diag}(G_k)^{-1/2}$, where G_k is a matrix equal to the sum of the first k iterations of the outer products of the gradient, and $\mathrm{diag}(\cdot)$ maps a matrix to its diagonal version.
- Iterate until convergence.

Results

The three algorithms have different stopping criteria. We run each for 100 s for parity. A summary of results is given in Table 2. Model fitting plots and algorithmic traceplots are shown in Fig. 5.

Table 2 Summary of results: last two rows show the average for the corresponding parameter (in horizontal direction) and algorithm (in vertical direction), after burn-in (the number in brackets is the corresponding standard deviation)

	MCMC	MC-CAVI	BBVI
Iterations	No. iterations = 2500 Burn-in = 1250	No. iterations = 300 $N = 10$ Burn-in = 150	No. iterations = 100 $N = 10$
ϑ	5.927 (0.117)	5.951 (0.009)	6.083 (0.476)
θ	1.248 (0.272)	8.880 (0.515)	0.442 (0.172)

All algorithms were executed for 10^2 s. The first row gives some algorithmic details

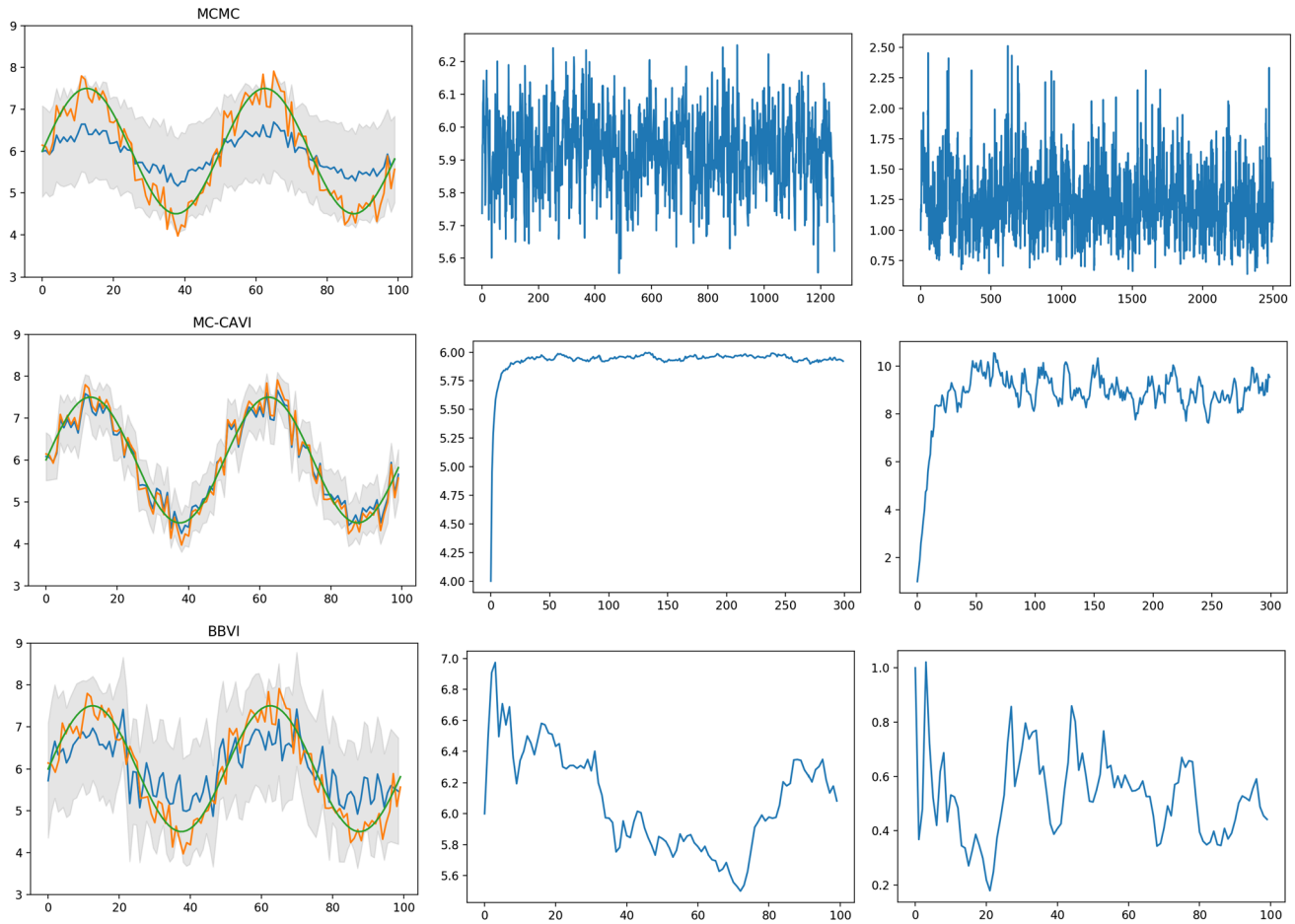

Fig. 5 Model fit (left panel), traceplots of ϑ (middle panel) and traceplots of θ (right panel) for the three algorithms: MCMC (first row), MC-CAVI (second row) and BBVI (third row)—for Example Model 2—when allowed 100 s of execution. In the plots showing model fit, the green line represents the data without noise, the orange line the data with noise; the blue line shows the corresponding posterior means and the grey area the pointwise 95% posterior credible intervals. (Color figure online)

Table 2 indicates that all three algorithms approximate the posterior mean of ϑ effectively; the estimate from MC-CAVI has smaller variability than the one of BBVI; the opposite holds for the variability in the estimates for θ. Figure 5 shows that the traceplots for BBVI are unstable, a sign that the gradient estimates have high variability. In contrast, MCMC and MC-CAVI perform rather well. Figure 6 shows the 'true' posterior density of ϑ (obtained from an expensive MCMC with 10,000 iterations—5000 burn-in) and the corresponding approximation obtained via MC-CAVI. In this case, the variational approximation is quite accurate at the estimation of the mean but underestimates the posterior variance (rather typically for a VI method). We mention that for BBVI we also tried to use normal laws as variational distributions—as this is mainly the standard choice in the literature—however, in this case, the performance of BBVI deteriorated even further.

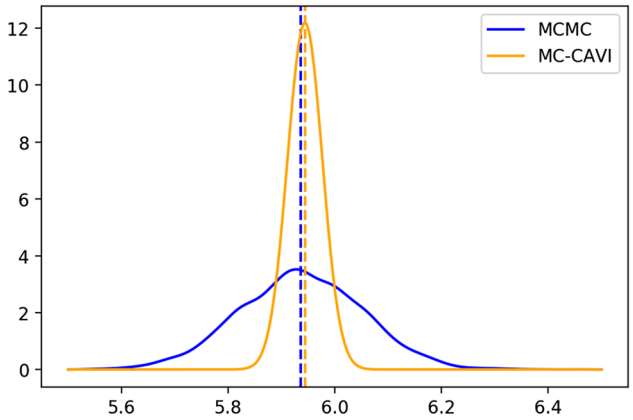

Fig. 6 Density plots for the true posterior of ϑ (blue line)—obtained via an expensive MCMC—and the corresponding approximate distribution provided by MC-CAVI. (Color figure online)

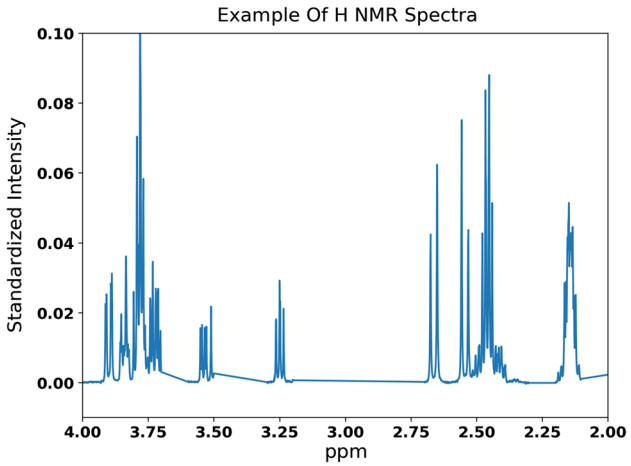

Fig. 7 An Example of ^{1}H NMR spectrun

4 Application to ^{1}H NMR spectroscopy

We demonstrate the utility of MC-CAVI in a statistical model proposed in the field of metabolomics by Astle et al. (2012), and used in NMR (Nuclear Magnetic Resonance) data analysis. Proton nuclear magnetic resonance (^{1}H NMR) is an extensively used technique for measuring abundance (concentration) of a number of metabolites in complex biofluids. NMR spectra are widely used in metabolomics to obtain profiles of metabolites present in biofluids. The NMR spectrum can contain information for a few hundreds of compounds. Resonance peaks generated by each compound must be identified in the spectrum after deconvolution. The spectral signature of a compound is given by a combination of peaks not necessarily close to each other. Such compounds can generate hundreds of resonance peaks, many of which overlap. This causes difficulty in peak identification and deconvolution. The analysis of NMR spectrum is further complicated by fluctuations in peak positions among spectra induced by uncontrollable variations in experimental conditions and the chemical properties of the biological samples, e.g. by the pH. Nevertheless, extensive information on the patterns of spectral resonance generated by human metabolites is now available in online databases. By incorporating this information into a Bayesian model, we can deconvolve resonance peaks from a spectrum and obtain explicit concentration estimates for the corresponding metabolites. Spectral resonances that cannot be deconvolved in this way may also be of scientific interest; these are modelled in Astle et al. (2012) using wavelet basis functions. More specifically, an NMR spectrum is a collection of peaks convoluted with various horizontal translations and vertical scalings, with each peak having the form of a Lorentzian curve. A number of metabolites of interest have known NMR spectrum shape, with the height of the peaks or their width in a particular experiment providing information about the abundance of each metabolite.

The zero-centred, standardized Lorentzian function is defined as:

$$\ell_\gamma(x) = \frac{2}{\pi}\frac{\gamma}{4x^2+\gamma^2} \tag{14}$$

where γ is the peak width at half height. An example of ^{1}H NMR spectrum is shown in Fig. 7. The x-axis of the spectrum measures chemical shift in parts per million (ppm) and corresponds to the resonance frequency. The y-axis measures relative resonance intensity. Each spectrum peak corresponds to magnetic nuclei resonating at a particular frequency in the biological mixture, with every metabolite having a characteristic molecular ^{1}H NMR 'signature'; the result is a convolution of Lorentzian peaks that appear in specific positions in ^{1}H NMR spectra. Each metabolite in the experiment usually gives rise to more than a 'multiplet' in the spectrum—i.e. linear combination of Lorentzian functions, symmetric around a central point. Spectral signature (i.e. pattern multiplets) of many metabolites are stored in public databases. The aim of the analysis is: (i) to deconvolve resonance peak in the spectrum and assign them to a particular metabolite; (ii) estimate the abundance of the catalogued metabolites; (iii) model the component of a spectrum that cannot be assigned to known compounds. Astle et al. (2012) propose a two-component joint model for a spectrum, in which the metabolites whose peaks we wish to assign explicitly are modelled parametrically, using information from the online databases, while the unassigned spectrum is modelled using wavelets.

4.1 The model

We now describe the model of Astle et al. (2012). The available data are represented by the pair $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ is a vector of n ordered points (of the order $10^3 - 10^4$) on the chemical

shift axis—often regularly spaced—and $\mathbf{y}$ is the vector of the corresponding resonance intensity measurements (scaled, so that they sum up to 1). The conditional law of $\mathbf{y}|\mathbf{x}$ is modelled under the assumption that $y_i|\mathbf{x}$ are independent normal variables and,

$$\mathbb{E}[\, y_i \,|\, \mathbf{x}\,] = \phi(x_i) + \xi(x_i), \quad 1 \le i \le n. \tag{15}$$

Here, the ϕ component of the model represents signatures that we wish to assign to target metabolites. The ξ component models signatures of remaining metabolites present in the spectrum, but not explicitly modelled. We refer to this latter as residual spectrum and we highlight the fact that it is important to account for it as it can unveil important information not captured by $\phi(\cdot)$. Function ϕ is constructed parametrically using results from the physical theory of NMR and information available online databases or expert knowledge, while ξ is modelled semiparametrically with wavelets generated by a mother wavelet (symlet 6) that resembles the Lorentzian curve.

More analytically,

$$\phi(x_i) = \sum_{m=1}^{M} t_m(x_i)\beta_m$$

where M is the number of metabolites modelled explicitly and $\beta = (\beta_1, \ldots, \beta_M)^\top$ is a parameter vector corresponding to metabolite concentrations. Function $t_m(\cdot)$ represents a continuous template function that specifies the NMR signature of metabolite m and it is defined as,

$$t_m(\delta) = \sum_{u} \sum_{v=1}^{V_{m,u}} z_{m,u}\, \omega_{m,u,v}\, \ell_\gamma(\delta - \delta^*_{m,u} - c_{m,u,v}), \quad \delta > 0, \tag{16}$$

where u is an index running over all multiplets assigned to metabolite m, v is an index representing a peak in a multiplet and $V_{m,u}$ is the number of peaks in multiplet u of metabolite m. In addition, $\delta^*_{m,u}$ specifies the theoretical position on the chemical shift axis of the centre of mass of the uth multiplet of the mth metabolite; $z_{m,u}$ is a positive quantity, usually equal to the number of protons in a molecule of metabolite m that contributes to the resonance signal of multiplet u; $\omega_{m,u,v}$ is the weight determining the relative heights of the peaks of the multiplet; $c_{m,u,v}$ is the translation determining the horizontal offsets of the peaks from the centre of mass of the multiplet. Both $\omega_{m,u,v}$ and $c_{m,u,v}$ can be computed by empirical estimates of the so-called J-coupling constants; see Hore (2015) for more details. The $z_{m,u}$'s and J-coupling constants information can be found in online databases or from expert knowledge.

The residual spectrum is modelled through wavelets,

$$\xi(x_i) = \sum_{j,k} \varphi_{j,k}(x_i)\vartheta_{j,k}$$

where $\varphi_{j,k}(\cdot)$ denote the orthogonal wavelet functions generated by the symlet-6 mother wavelet, see Astle et al. (2012) for full details; here, $\vartheta = (\vartheta_{1,1}, \ldots, \vartheta_{j,k}, \ldots)^\top$ is the vector of wavelet coefficients. Indices j, k correspond to the kth wavelet in the jth scaling level.

Finally, overall, the model for an NMR spectrum can be re-written in matrix form as:

$$\mathcal{W}(\mathbf{y} - \mathbf{T}\beta) = \mathbf{I}_{n_1}\vartheta + \epsilon, \quad \boldsymbol{\epsilon} \sim \mathrm{N}(0, \mathbf{I}_{n_1}/\theta), \tag{17}$$

where $\mathcal{W} \in \mathbb{R}^{n \times n_1}$ is the inverse wavelet transform, M is the total number of known metabolites, $\mathbf{T}$ is an $n \times M$ matrix with its (i, m)th entry equal to $t_m(x_i)$ and θ is a scalar precision parameter.

4.2 Prior specification

Astle et al. (2012) assign the following prior distribution to the parameters in the Bayesian model. For the concentration parameters, we assume

$$\beta_m \sim \mathrm{TN}(e_m, 1/s_m, 0, \infty),$$

where $e_m = 0$ and $s_m = 10^{-3}$, for all $m = 1, \ldots, M$. Moreover,

$$\gamma \sim \mathrm{LN}(0, 1);$$
$$\delta^*_{m,u} \sim \mathrm{TN}(\hat{\delta}^*_{m,u}, 10^{-4}, \hat{\delta}^*_{m,u} - 0.03, \hat{\delta}^*_{m,u} + 0.03),$$

where LN denotes a log-normal distribution and $\hat{\delta}^*_{m,u}$ is the estimate for $\delta^*_{m,u}$ obtained from the online database HMDB (see Wishart et al. 2007, 2008, 2012, 2017). In the regions of the spectrum where both parametric (i.e. ϕ) and semiparametric (i.e. ξ) components need to be fitted, the likelihood is unidentifiable. To tackle this problem, Astle et al. (2012) opt for shrinkage priors for the wavelet coefficients and include a vector of hyperparameters ψ—each component $\psi_{j,k}$ of which corresponds to a wavelet coefficient—to penalize the semiparametric component. To reflect prior knowledge that NMR spectra are usually restricted to the half plane above the chemical shift axis, Astle et al. (2012) introduce a vector of hyperparameters τ, each component of which, τ_i, corresponds to a spectral data point, to further penalize spectral reconstructions in which some components of $\mathcal{W}^{-1}\boldsymbol{\vartheta}$ are less than a small negative threshold. In conclusion, Astle

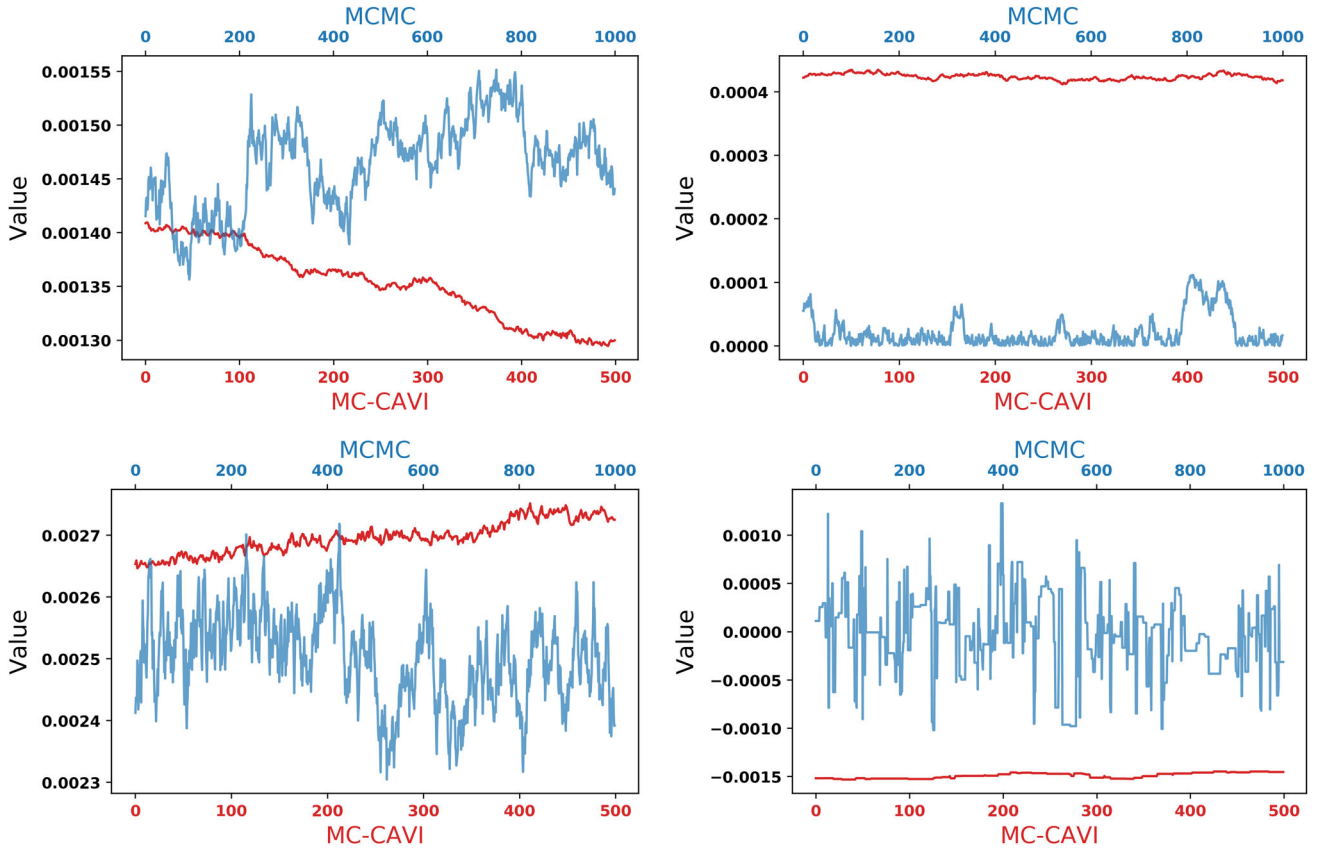

Fig. 8 Traceplots of parameter value against number of iterations after the burn-in period for β_3 (upper left panel), β_4 (upper right panel), β_9 (lower left panel) and $\delta_{4,1}$ (lower right panel). The y-axis corresponds to the obtained parameter values (the mean of the distribution q for MC-CAVI and traceplots for MCMC). The red line shows the results from MC-CAVI and the blue line from MCMC. Both algorithms are executed for the same (approximately) amount of time. (Color figure online)

et al. (2012) specify the following joint prior density for $(\vartheta, \psi, \tau, \theta)$,

$$p(\vartheta, \psi, \tau, \theta) \propto \theta^{a+\frac{n+n_1}{2}-1} \left\{ \prod_{j,k} \psi_{j,k}^{c_j-0.5} \exp\left(-\tfrac{\psi_{j,k} d_j}{2}\right) \right\} \times \exp\left\{ -\tfrac{\theta}{2} \left(e + \sum_{j,k} \psi_{j,k}\, \vartheta_{j,k}^2 + r \sum_{i=1}^{n} (\tau_i - h)^2 \right) \right\} \times \mathbb{1}\left\{ \mathcal{W}^{-1}\vartheta \geq \tau,\ h\mathbf{1}_n \geq \tau \right\},$$

where ψ introduces local shrinkage for the marginal prior of ϑ and τ is a vector of n truncation limits, which bounds $\mathcal{W}^{-1}\vartheta$ from below. The truncation imposes an identifiability constraint: without it, when the signature template does not match the shape of the spectral data, the mismatch will be compensated by negative wavelet coefficients, such that an ideal overall model fit is achieved even though the signature template is erroneously assigned and the concentration of metabolites is overestimated. Finally we set $c_j = 0.05, d_j = 10^{-8}, h = -0.002, r = 10^5, a = 10^{-9}, e = 10^{-6}$; see Astle et al. (2012) for more details.

4.3 Results

BATMAN is an R package for estimating metabolite concentrations from NMR spectral data using a specifically designed MCMC algorithm (Hao et al. 2012) to perform posterior inference from the Bayesian model described above. We implement a MC-CAVI version of BATMAN and compare its performance with the original MCMC algorithm. Details of the implementation of MC-CAVI are given in "Appendix D". Due to the complexity of the model and the data size, it is challenging for both algorithms to reach convergence. We run the two methods, MC-CAVI and MCMC, for approximately an equal amount of time, to analyse a full spectrum with 1530 data points and modelling parametrically 10 metabolites. We fix the number of iterations for MC-CAVI to 1000, with a burn-in of 500 iterations; we set the Monte Carlo size to $N = 10$ for all iterations. The execution time for this MC-CAVI algorithms is 2048 s. For the MCMC algorithm, we fix the number of iterations to 2000, with a burn-in of 1000 iterations. This MCMC algorithm has an execution time of 2098 s.

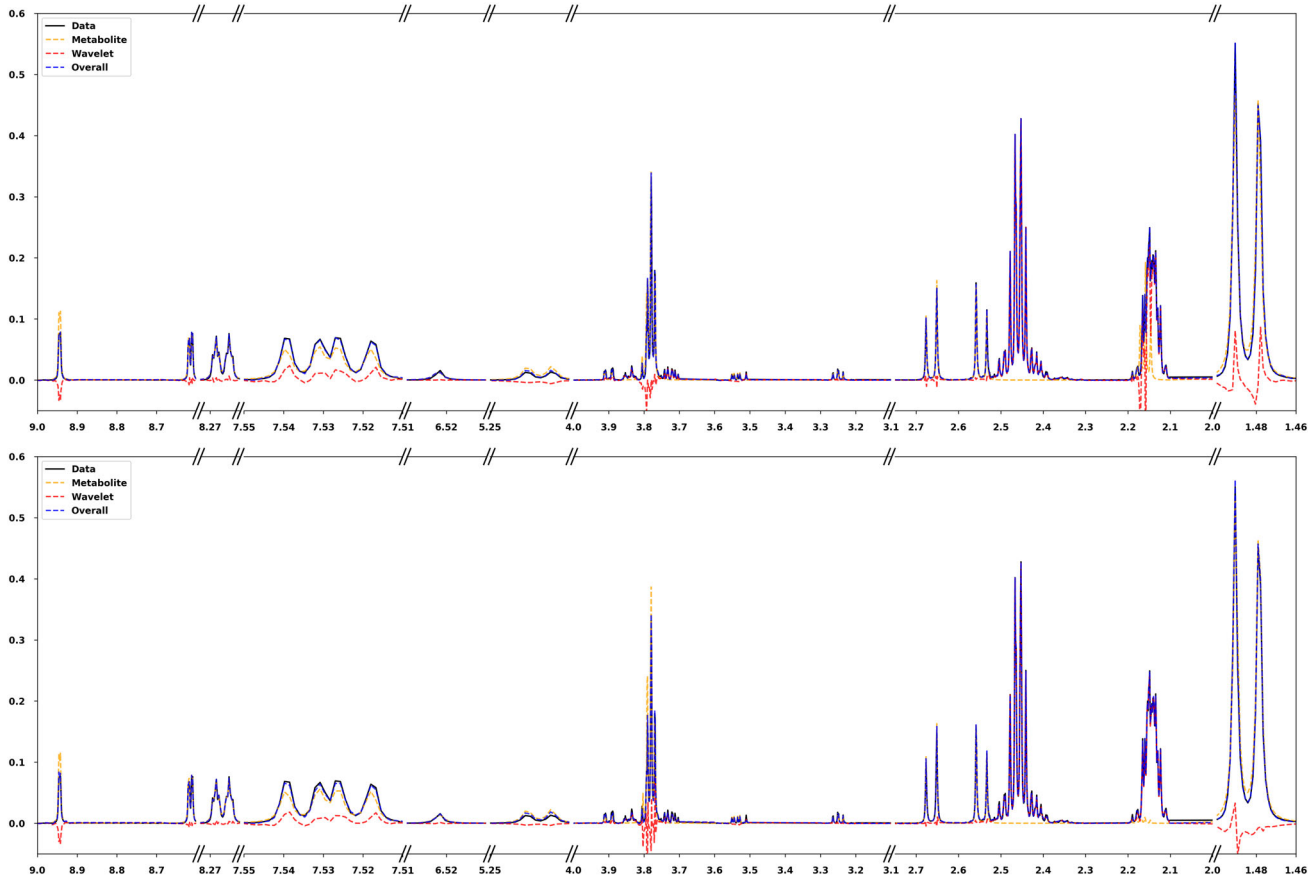

Fig. 9 Comparison of MC-CAVI and MCMC in terms of spectral fit. The upper panel shows the Spectral Fit from MC-CAVI algorithm. The lower panel shows the Spectral Fit from MCMC algorithm. The x-axis corresponds to chemical shift measure in ppm. The y-axis corresponds to standard density

Table 3 Estimation of β obtained with MC-CAVI and MCMC

	β_1	β_2	$\boldsymbol{\beta_3}$	$\boldsymbol{\beta_4}$	β_5
MC-CAVI					
Mean	6.0e−6	7.8e−5	1.4e−3	4.2e−4	2.6e−5
SD	1.8e−11	4.0e−11	1.3e−11	1.0e−11	6.2e−11
MCMC					
Mean	1.2e−5	4.0e−5	1.5e−3	2.1e−5	3.4e−5
SD	1.1e−10	5.0e−10	1.6e-9	6.4e−10	3.9e−10
	β_6	β_7	β_8	$\boldsymbol{\beta_9}$	β_{10}
MC-CAVI					
Mean	6.1e−4	3.0e−5	1.9e−4	2.7e−3	1.0e−3
SD	1.5e−11	1.6e−11	3.9e−11	1.6e−11	3.6e−11
MCMC					
Mean	6.0e−4	3.0e−5	1.8e−4	2.5e−3	1.0e−3
SD	2.3e−10	7.5e−11	3.7e−10	5.1e-9	7.9e−10

The coefficients of β for which the posterior means obtained with the two algorithms differ by more than 1.0e−4 are shown in bold

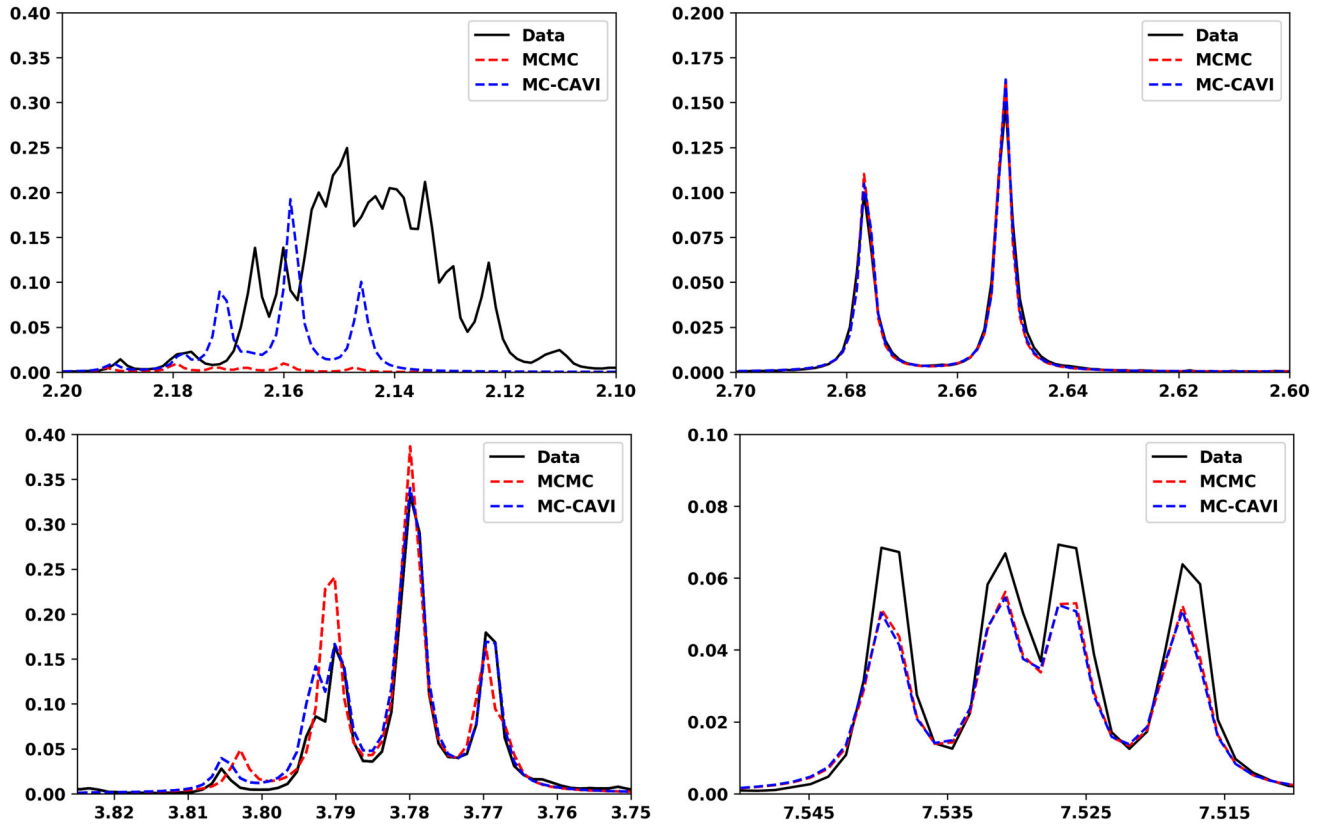

Fig. 10 Comparison of metabolites fit obtained with MC-CAVI and MCMC. The x-axis corresponds to chemical shift measure in ppm. The y-axis corresponds to standard density. The upper left panel shows areas around ppm value 2.14 (β_4 and β_9). The upper right panel shows areas around ppm 2.66 (β_6). The lower left panel shows areas around ppm value 3.78 (β_3 and β_9). The lower right panel shows areas around ppm 7.53 (β_{10})

In ^{1}H NMR analysis, β (the concentration of metabolites in the biofluid) and $\delta^*_{m,u}$ (the peak positions) are the most important parameters from a scientific point of view. Traceplots of four examples (β_3, β_4, β_9 and $\delta_{4,1}$) are shown in Fig. 8. These four parameters are chosen due to the different performance of the two methods, which are closely examined in Fig. 10. For β_3 and β_9, traceplots are still far from convergence for MCMC, while they move toward the correct direction (see Fig. 8) when using MC-CAVI. For β_4 and $\delta_{4,1}$, both parameters reach a stable regime very quickly in MC-CAVI, whereas the same parameters only make local moves when implementing MCMC. For the remaining parameters in the model, both algorithms present similar results.

Figure 9 shows the fit obtained from both the algorithms, while Table 3 reports posterior estimates for β. From Fig. 9, it is evident that the overall performance of MC-CAVI is similar as that of MCMC since in most areas, the metabolites fit (orange line) captures the shape of the original spectrum quite well. Table 3 shows that, similar to standard VI behaviour, MC-CAVI underestimates the variance of the posterior density. We examine in more detail the posterior distribution of the β coefficients for which the posterior means obtained with the two algorithms differ more than 1.0e−4. Figure 10 shows that MC-CAVI manages to capture the shapes of the peaks while MCMC does not, around ppm values of 2.14 and 3.78, which correspond to spectral regions where many peaks overlap making peak deconvolution challenging. This is probably due to the faster convergence of MC-CAVI. Figure 10 shows that for areas with no overlapping (e.g. around ppm values of 2.66 and 7.53), MC-CAVI and MCMC produce similar results.

Comparing MC-CAVI and MCMC's performance in the case of the NMR model, we can draw the following conclusions:

- In NMR analysis, if many peaks overlap (see Fig. 10), MC-CAVI can provide better results than MCMC.
- In high-dimensional models, where the number of parameters grows with the size of data, MC-CAVI can converge faster than MCMC.
- Choice of N is important for optimising the performance of MC-CAVI. Building on results derived for other Monte Carlo methods (e.g. MCEM), it is reasonable to choose a

relatively small number of Monte Carlo iterations at the beginning when the algorithm can be far from regions of parameter space of high posterior probability, and gradually increase the number of Monte Carlo iterations, with the maximum number taken once the algorithm has reached a mode.

5 Discussion

As a combination of VI and MCMC, MC-CAVI provides a powerful inferential tool particularly in high dimensional settings when full posterior inference is computationally demanding and the application of optimization and of noisy-gradient-based approaches, e.g. BBVI, is hindered by the presence of hard constraints. The MCMC step of MC-CAVI is necessary to deal with parameters for which VI approximation distributions are difficult or impossible to derive, for example due to the impossibility to derive closed-form expression for the normalising constant. General Monte Carlo algorithms such as sequential Monte Carlo and Hamiltonian Monte Carlo can be incorporated within MC-CAVI. Compared with MCMC, the VI step of MC-CAVI speeds up convergence and provides reliable estimates in a shorter time. Moreover, MC-CAVI scales better in high-dimensional settings. As an optimization algorithm, MC-CAVI's convergence monitoring is easier than MCMC. Moreover, MC-CAVI offers a flexible alternative to BBVI. This latter algorithm, although very general and suitable for a large range of complex models, depends crucially on the quality of the approximation to the true target provided by the variational distribution, which in high dimensional setting (in particular with hard constraints) is very difficult to assess.

Acknowledgements We thank two anonymous referees for their comments that greatly improved the content of the paper. AB acknowledges funding by the Leverhulme Trust Prize.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

A Proof of Lemma 1

Proof Part (i): For a neighborhood of λ^*, we can chose a sub-neighborhood V as described in Assumption 3. For some small $\epsilon > 0$, the set $V_0 = \{\lambda : \text{ELBO}(q(\lambda)) \geq \text{ELBO}(q(\lambda^*)) - \epsilon\}$ has a connected component, say V', so that $\lambda^* \in V'$ and $V' \subseteq V$; we can assume that V' is compact. Assumption 3 implies that $M(V') \subseteq V_0$; in fact, since $M(V')$ is connected and contains λ^*, we have $M(V') \subseteq V'$. This completes the proof of part (i) of Definition 1.
Part (ii): Let $\lambda \in V'$. Consider the sequence $\{M^k(\lambda)\}_k$ with a convergent subsequence, $M^{a_k}(\lambda) \to \lambda_1 \in V'$, for increasing integers $\{a_k\}$. Thus, we have that the following holds, $\text{ELBO}(q(M^{a_{k+1}}(\lambda))) \geq \text{ELBO}(q(M(M^{a_k}(\lambda)))) \to \text{ELBO}(q(M(\lambda_1)))$, whereas we also have that $\text{ELBO}(q(M^{a_{k+1}}(\lambda))) \to \text{ELBO}(q(\lambda_1))$. These two last limits give the implication that $\text{ELBO}(q(M(\lambda_1))) = \text{ELBO}(q(\lambda_1))$, so that $\lambda_1 = \lambda^*$. We have shown that any convergent subsequence of $\{M^k(\lambda)\}_k$ has limit λ^*; the compactness of V' gives that also $M^k(\lambda) \to \lambda^*$. This completes the proof of part (ii) of Definition 1. □

B Proof of Theorem 1

Proof Let V_1 be as V' within the proof of Lemma 1. Define $V_2 = \{\lambda \in V_1 : |\lambda - \lambda^*| \geq \epsilon\}$, for an $\epsilon > 0$ small enough so that $V_1 \neq \emptyset$. For $\lambda \in V_2$, we have $M(\lambda) \neq \lambda$, thus there are $\nu, \nu_1 > 0$ such that for all $\lambda \in V_2$ and for all λ' with $|\lambda' - M(\lambda)| < \nu$, we obtain that $\text{ELBO}(q(\lambda')) - \text{ELBO}(q(\lambda)) > \nu_1$. Also, due to continuity and compactness, there is $\nu_2 > 0$ such that for all $\lambda \in V_1$ and for all λ' such that $|\lambda' - M(\lambda)| < \nu_2$, we have $\lambda' \in V_1$. Let $R = \sup_{\lambda,\lambda' \in V_1}\{\text{ELBO}(q(\lambda)) - \text{ELBO}(q(\lambda'))\}$ and $k_0 = [R/\nu_1]$ where $[\cdot]$ denotes integer part. Notice that given $\lambda_N^k := \mathcal{M}_N^k(\lambda)$, we have that $\{|\mathcal{M}_N^{k+1} - M(\lambda_N^k)| < \nu_2\} \subseteq \{\lambda_N^{k+1} \in V_1\}$. Consider the event $F_N = \{\lambda_N^k \in V_1\,;\, k = 0, \ldots, k_0\}$. Under Assumption 4, we have that $\text{Prob}[F_N] \geq p^{k_0}$ for p arbitrarily close to 1. Within F_N, we have that $|\lambda_N^k - \lambda^*| < \epsilon$ for some $k \leq k_0$, or else $\lambda_N^k \in V_2$ for all $k \leq k_0$, giving that $\text{ELBO}(q(\lambda_N^k)) - \text{ELBO}(q(\lambda)) > \nu_1 \cdot k_0 > R$, which is impossible. □

C Gradient expressions for BBVI

$$
\begin{aligned}
\nabla_{\alpha_\vartheta} \log q(\vartheta) &= (\vartheta - \alpha_\vartheta) \cdot \exp(-\gamma_\vartheta),\\
\nabla_{\gamma_\vartheta} \log q(\vartheta) &= -\tfrac{1}{2} + \tfrac{(\vartheta-\alpha_\vartheta)^2}{2} \cdot \exp(-\gamma_\vartheta),\\
\nabla_{\alpha_\theta} \log q(\theta) &= \left(\gamma_\theta - \tfrac{\Gamma'(\exp(\alpha_\theta))}{\Gamma(\exp(\alpha_\theta))} + \log(\theta)\right) \cdot \exp(\alpha_\theta),\\
\nabla_{\gamma_\theta} \log q(\theta) &= \exp(\alpha_\theta) - \theta \cdot \exp(\gamma_\theta),
\end{aligned}
$$

$$\nabla_{\alpha_{\kappa_j}} \log q(\kappa_j, \psi_j) = \frac{\kappa_j - \alpha_{\kappa_j}}{\exp(2\gamma_{\kappa_j})} + \frac{\phi(\frac{\psi_j - \alpha_{\kappa_j}}{\exp(\gamma_{\kappa_j})}) - \phi(\frac{-\psi_j - \alpha_{\kappa_j}}{\exp(\gamma_{\kappa_j})})}{\exp(\gamma_{\kappa_j})(\Phi(\frac{\psi_j - \alpha_{\kappa_j}}{\exp(\gamma_{\kappa_j})}) - \Phi(\frac{-\psi_j - \alpha_{\kappa_j}}{\exp(\gamma_{\kappa_j})}))},$$

$$1 \le j \le n$$

$$\nabla_{\alpha_{\psi_j}} \log q(\kappa_j, \psi_j) = \frac{\psi_j - \alpha_{\psi_j}}{\exp(2\gamma_{\psi_j})} + \frac{\phi(\frac{2-\alpha_{\psi_j}}{\exp(\gamma_{\psi_j})}) - \phi(\frac{-\alpha_{\psi_j}}{\exp(\gamma_{\psi_j})})}{\exp(\gamma_{\psi_j})(\Phi(\frac{2-\alpha_{\psi_j}}{\exp(\gamma_{\psi_j})}) - \Phi(\frac{-\alpha_{\psi_j}}{\exp(\gamma_{\psi_j})}))},$$

$$1 \le j \le n$$

$$\nabla_{\gamma_{\kappa_j}} \log q(\kappa_j, \psi_j) = \frac{(\kappa_j - \alpha_{\kappa_j})^2}{\exp(2\gamma_{\kappa_j})} - 1 + \frac{(\psi_j - \alpha_{\kappa_j})\phi(\frac{\psi_j - \alpha_{\kappa_j}}{\exp(\gamma_{\kappa_j})}) + (\psi_j + \alpha_{\kappa_j})\phi(\frac{-\psi_j - \alpha_{\kappa_j}}{\exp(\gamma_{\kappa_j})})}{\exp(\gamma_{\kappa_j})(\Phi(\frac{\psi_j - \alpha_{\kappa_j}}{\exp(\gamma_{\kappa_j})}) - \Phi(\frac{-\psi_j - \alpha_{\kappa_j}}{\exp(\gamma_{\kappa_j})}))},$$

$$1 \le j \le n$$

$$\nabla_{\gamma_{\psi_j}} \log q(\kappa_j, \psi_j) = \frac{(\psi_j - \alpha_{\psi_j})^2}{\exp(2\gamma_{\psi_j})} - 1 + \frac{(2 - \alpha_{\psi_j})\boldsymbol{\phi}(\frac{2-\alpha_{\psi_j}}{\exp(\gamma_{\psi_j})}) + (\alpha_{\psi_j})\boldsymbol{\phi}(\frac{-\alpha_{\psi_j}}{\exp(\gamma_{\psi_j})})}{\exp(\gamma_{\psi_j})(\Phi(\frac{2-\alpha_{\psi_j}}{\exp(\gamma_{\psi_j})}) - \Phi(\frac{-\alpha_{\psi_j}}{\exp(\gamma_{\psi_j})}))},$$

$$1 \le j \le n.$$

D MC-CAVI implementation of BATMAN

In the MC-CAVI implementation of BATMAN, taking both computation efficiency and model structure into consideration, we assume that the variational distribution factorises over four partitions of the parameter vectors, $q(\beta, \delta^*, \gamma)$, $q(\vartheta, \tau)$, $q(\psi)$, $q(\theta)$. This factorization is motivated by the original Metropolis–Hastings block updates in Astle et al. (2012). Let B denote the wavelet basis matrix defined by the transform $\mathcal{W}$, so $\mathcal{W}(B) = \mathbf{I}_{n_1}$. We use v_{-i} to represent vector v without the ith component and analogous notation for matrices (resp., without the ith column).
Set $\mathbb{E}(\theta) = 2a/e$, $\mathbb{E}(\vartheta_{j,k}^2) = 0$, $\mathbb{E}(\vartheta) = 0$, $\mathbb{E}(\tau) = 0$, $\mathbb{E}(\mathbf{T}\beta) = \mathbf{y}$, $\mathbb{E}\big((\mathbf{T}\beta)^\top(\mathbf{T}\beta)\big) = \mathbf{y}^\top\mathbf{y}$.
For each iteration:

1. Set $q(\psi_{j,k}) = \text{Gamma}\big(c_j + \frac{1}{2}, \frac{\mathbb{E}(\theta)\mathbb{E}(\vartheta_{j,k}^2) + d_j}{2}\big)$; calculate $\mathbb{E}(\psi_{j,k})$.
2. Set $q(\theta) = \text{Gamma}(c, c')$, where we have defined,

$$c = a_1 + n_1 + \frac{n}{2},$$

$$c' = \frac{1}{2}\Bigg\{\sum_{j,k}\mathbb{E}(\psi_{j,k})\mathbb{E}(\vartheta_{j,k}^2) + \mathbb{E}\big((\mathcal{W}\mathbf{y} - \mathcal{W}\mathbf{T}\beta - \vartheta)^\top(\mathcal{W}\mathbf{y} - \mathcal{W}\mathbf{T}\beta - \vartheta)\big) + r(\mathbb{E}(\tau) - h\mathbf{1}_n) + e\Bigg\};$$

calculate $\mathbb{E}(\theta)$.
3. Use Monte Carlo to draw N samples from $q(\beta, \delta^*_{m,u}, \gamma)$, which is derived via (4) as,

$$q(\beta, \delta^*, \gamma) \propto \exp\Big\{-\frac{\mathbb{E}(\theta)}{2}\big((\mathcal{W}\boldsymbol{T}\beta)^\top\mathcal{W}\boldsymbol{T}\beta - 2\mathcal{W}\boldsymbol{T}\beta(\mathcal{W}\mathbf{y} - \mathbb{E}(\vartheta))\big)\Big\} \times p(\beta)p(\delta^*)p(\gamma),$$

where $p(\beta)$, $p(\delta^*)$, $p(\gamma)$ are the prior distributions specified in Sect. 4.2.

- Use a Gibbs sampler update to draw samples from $q(\beta|\delta^*_{m,u}, \gamma)$. Draw each component of $\beta = (\beta_m)$ from a univariate normal, truncated below at zero, with precision and mean parameters given, respectively, by

$$P := s_m + \mathbb{E}(\theta)(\mathcal{W}\boldsymbol{T}_i)^\top(\mathcal{W}\boldsymbol{T}_i),$$
$$(\mathcal{W}\boldsymbol{T}_i)^\top(\mathcal{W}\mathbf{y} - \mathcal{W}\boldsymbol{T}_{-i}\beta_{-i} - \mathbb{E}(\vartheta))\mathbb{E}(\theta)/P.$$

- Use Metropolis–Hastings to update γ. Propose $\log(\gamma') \sim \text{N}(\log(\gamma), V_\gamma^2)$. Perform accept/reject. Adapt V_γ^2 to obtain average acceptance rate of approximately 0.45.
- Use Metropolis–Hastings to update $\delta^*_{m,u}$. Propose,

$$(\delta^*_{m,u})' \sim \text{TN}(\delta^*_{m,u}, V^2_{\delta^*_{m,u}}, \hat{\delta}^*_{m,u} - 0.03, \hat{\delta}^*_{m,u} + 0.03).$$

Perform accept/reject. Adapt $V^2_{\delta^*_{m,u}}$ to target acceptance rate 0.45.

Calculate $\mathbb{E}(\mathbf{T}\beta)$ and $\mathbb{E}\big((\mathbf{T}\beta)^\top(\mathbf{T}\beta)\big)$.
4. Use Monte Carlo to draw N samples from $q(\vartheta, \tau)$, which is derived via (4) as,

$$q(\vartheta, \tau) \propto \exp\Big\{-\frac{\mathbb{E}(\theta)}{2}\Big(\sum_{j,k}\vartheta_{j,k}\big((\psi_{j,k} + 1)\,\vartheta_{j,k} - 2\big(\mathcal{W}\mathbf{y} - \mathcal{W}\mathbb{E}(\boldsymbol{T}\beta)\big)_{j,k}\big) + r\sum_{i=1}^{n}(\tau_i - h)^2\Big)\Big\} \times \mathbb{I}\big\{\mathcal{W}^{-1}\vartheta \ge \tau,\ h\mathbf{1}_n \ge \tau\big\}$$

- Use Gibbs sampler to draw from $q(\vartheta|\tau)$. Draw $\vartheta_{j,k}$ from:

$$\text{TN}\Big(\frac{1}{1+\mathbb{E}(\psi_{j,k})}\big(\mathcal{W}\mathbf{y} - \mathcal{W}\mathbb{E}(\boldsymbol{T}\beta)\big)_{j,k}, \frac{1}{\mathbb{E}(\theta)(1+\mathbb{E}(\psi_{j,k}))}, L, U\Big)$$

where we have set,

$$L = \max_{i:B_{i\{j,k\}}>0} \frac{\tau_i - B_{i-\{j,k\}}\vartheta_{-\{j,k\}}}{B_{i\{j,k\}}}$$

$$U = \min_{i:B_{i\{j,k\}}<0} \frac{\tau_i - B_{i-\{j,k\}}\vartheta_{-\{j,k\}}}{B_{i\{j,k\}}}$$

and $B_{i\{j,k\}}$ is the (j,k)th element of the ith column of B.

- Use Gibbs sampler to update τ_i. Draw,

$$\tau_i \sim \mathrm{TN}\big(h, 1/(\mathbb{E}(\theta)r), -\infty, \min\left\{h, (\mathcal{W}^{-1}\vartheta)_i\right\}\big).$$

Calculate $\mathbb{E}(\vartheta^2_{j,k})$, $\mathbb{E}(\vartheta)$, $\mathbb{E}(\tau)$.

References

Astle, W., De Iorio, M., Richardson, S., Stephens, D., Ebbels, T.: A Bayesian model of NMR spectra for the deconvolution and quantification of metabolites in complex biological mixtures. J. Am. Stat. Assoc. **107**(500), 1259–1271 (2012)

Beaumont, M.A., Zhang, W., Balding, D.J.: Approximate Bayesian computation in population genetics. Genetics **162**(4), 2025–2035 (2002)

Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (1999)

Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)

Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)

Booth, J.G., Hobert, J.P.: Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. J. R. Stat. Soc. Ser. B (Statistical Methodology) **61**(1), 265–285 (1999)

Bottou, L.: Stochastic Gradient Descent Tricks, pp. 421–436. Springer, Berlin (2012)

Casella, G., Robert, C.P.: Rao–Blackwellisation of sampling schemes. Biometrika **83**(1), 81–94 (1996)

Chan, K., Ledolter, J.: Monte Carlo EM estimation for time series models involving counts. J. Am. Stat. Assoc. **90**(429), 242–252 (1995)

Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and Stochastic optimization. J. Mach. Learn. Res. **12**(Jul), 2121–2159 (2011)

Forbes, F., Fort, G.: Combining Monte Carlo and mean-field-like methods for inference in hidden Markov random fields. IEEE Trans. Image Process. **16**(3), 824–837 (2007)

Fort, G., Moulines, E., et al.: Convergence of the Monte Carlo expectation maximization for curved exponential families. Ann. Stat. **31**(4), 1220–1259 (2003)

Hao, J., Astle, W., De Iorio, M., Ebbels, T.M.: BATMAN—an R package for the automated quantification of metabolites from nuclear magnetic resonance spectra using a Bayesian model. Bioinformatics **28**(15), 2088–2090 (2012)

Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. J. Mach. Learn. Res. **14**(1), 1303–1347 (2013)

Hore, P.J.: Nuclear Magnetic Resonance. Oxford University Press, Oxford (2015)

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Mach. Learn. **37**(2), 183–233 (1999)

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. J. Mach. Learn. Res. **18**(1), 430–474 (2017)

Levine, R.A., Casella, G.: Implementations of the Monte Carlo EM algorithm. J. Comput. Graph. Stat. **10**(3), 422–439 (2001)

Ranganath, R., Gerrish, S., Blei, D.: Black box variational inference. Artif. Intell. Stat. **33**, 814–822 (2014)

Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**(3), 400–407 (1951)

Ross, S.M.: Simulation. Elsevier, Amsterdam (2002)

Sisson, S.A., Fan, Y., Tanaka, M.M.: Sequential Monte Carlo without likelihoods. Proc. Nat. Acad. Sci. **104**(6), 1760–1765 (2007)

Tran, M.-N., Nott, D.J., Kuk, A.Y., Kohn, R.: Parallel variational Bayes for large datasets with an application to generalized linear mixed models. J. Comput. Graph. Stat. **25**(2), 626–646 (2016)

Tran, M.-N., Nguyen, D.H., Nguyen, D.: Variational Bayes on Manifolds (2019). arXiv:1908.03097

Wainwright, M.J., Jordan, M.I., et al.: Graphical Models, Exponential Families, and Variational Inference, vol. 1. Now Publishers, Inc., Hanover (2008)

Wei, G.C., Tanner, M.A.: A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. J. Am. Stat. Assoc. **85**(411), 699–704 (1990)

Wishart, D.S., Tzur, D., Knox, C., Eisner, R., Guo, A.C., Young, N., Cheng, D., Jewell, K., Arndt, D., Sawhney, S., et al.: HMDB: the human metabolome database. Nucl. Acids Res. **35**(suppl1), D521–D526 (2007)

Wishart, D.S., Knox, C., Guo, A.C., Eisner, R., Young, N., Gautam, B., Hau, D.D., Psychogios, N., Dong, E., Bouatra, S., et al.: HMDB: a knowledgebase for the human metabolome. Nucl. Acids Res. **37**(suppl1), D603–D610 (2008)

Wishart, D.S., Jewison, T., Guo, A.C., Wilson, M., Knox, C., Liu, Y., Djoumbou, Y., Mandal, R., Aziat, F., Dong, E., et al.: HMDB 3.0—the human metabolome database in 2013. Nucl. Acids Res. **41**(1), D801–D807 (2012)

Wishart, D.S., Feunang, Y.D., Marcu, A., Guo, A.C., Liang, K., Vázquez-Fresno, R., Sajed, T., Johnson, D., Li, C., Karu, N., et al.: HMDB 4.0: the human metabolome database for 2018. Nucl. Acids Res. **46**(1), D608–D617 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.