JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2017, VOL. 112, NO. 517, 137–168, Theory and Methods
http://dx.doi.org/10.1080/01621459.2016.1197833

Taylor & Francis
Taylor & Francis Group

Fast Approximate Inference for Arbitrarily Large Semiparametric Regression Models via Message Passing

M. P. Wand

School of Mathematical and Physical Sciences, University of Technology Sydney, Sydney, Australia, and Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers, Queensland University of Technology (QUT), Brisbane, Australia

ABSTRACT

We show how the notion of *message passing* can be used to streamline the algebra and computer coding for fast approximate inference in large Bayesian semiparametric regression models. In particular, this approach is amenable to handling *arbitrarily large* models of particular types once a set of primitive operations is established. The approach is founded upon a message passing formulation of mean field variational Bayes that utilizes *factor graph* representations of statistical models. The underlying principles apply to general Bayesian hierarchical models although we focus on semiparametric regression. The notion of factor graph fragments is introduced and is shown to facilitate compartmentalization of the required algebra and coding. The resultant algorithms have ready-to-implement closed form expressions and allow a broad class of arbitrarily large semiparametric regression models to be handled. Ongoing software projects such as Infer.NET and Stan support variational-type inference for particular model classes. This article is not concerned with software packages *per se* and focuses on the underlying tenets of scalable variational inference algorithms. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received October 2016
Revised March 2016

KEYWORDS
Factor graphs; Generalized additive models; Generalized linear mixed models; Low-rank smoothing splines; Mean field variational Bayes; Scalable statistical methodology; Variational message passing

1. Introduction

We derive algorithmic primitives that afford fast approximate inference for arbitrarily large semiparametric regression models. The fit updating steps required for fitting a simple semiparametric regression model, such as Gaussian response nonparametric regression, can also be used for a much larger model involving, for example, multiple predictors, group-specific curves and non-Gaussian responses. Such update formulas only need to be derived and implemented once, representing enormous savings in terms of algebra and computing coding.

Semiparametric regression extends classical statistical models, such as generalized linear models and linear mixed models, to accommodate nonlinear predictor effects. The essence of the extension is penalization of basis functions such as B-splines and Daubechies wavelets. Such penalization can be achieved through random effects models that have the same form as those used traditionally in longitudinal and multilevel data analysis. Generalized additive models, group-specific curve models, and varying coefficient models are some of the families of models that are included within semiparametric regression. If a Bayesian approach is adopted then semiparametric regression can be couched within the directed acyclic graphical models infrastructure and, for example, Markov chain Monte Carlo (MCMC) and mean field variational Bayes (MFVB) algorithms and software can be used for fitting and inference. The MFVB approach has the advantage of being scalable to very large models and big datasets. Recent articles by the author that describe MCMC and MFVB approaches to semiparametric regression analysis include Ruppert, Wand and Carroll (2009), Wand (2009), Marley and Wand (2010), Wand and Ormerod (2011), and Luts, Broderick, and Wand (2014).

In this article, we revisit MFVB for semiparametric regression but instead work with an approach known as *variational message passing* (VMP) (Winn and Bishop 2005). The MFVB and VMP approaches each lead to ostensibly different iterative algorithms but, in a wide range of models, converge to the identical posterior density function approximations since they are each founded upon the same optimization problem. VMP has the advantage that its iterative updates are amenable to modularization, and extension to arbitrarily large models, via the notion of *factor graph fragments*. Factor graphs (Frey et al. 1998), described in Section 2.3, are a relatively new graphical concept. As explained in Minka (2005), mean field variational approximation iterative updates can be expressed as *messages* passed between nodes on a suitable factor graph. *Message passing* is a general principle in software engineering for efficient computing within so-called distributed systems (e.g., Ghosh 2015). In the contemporary statistics literature, Jordan (2004) explained how message passing can be used to streamline the computation of marginal probability mass functions of the nodes on large discrete random variable probabilistic undirected trees as a pedagogical special case of the factor graph treatment given in Kschischang, Frey, and Loeliger (2001). This particular message passing strategy is known as the *sum–product algorithm*. Despite its appeal for efficient and modular computation on large graphical models, message passing on factor graphs is not

CONTACT M. P. Wand matt.wand@uts.edu.au School of Mathematical and Physical Sciences, University of Technology Sydney, P.O. Box 123, Broadway 2007, Sydney, Australia.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

well known in mainstream statistics. The thrust of this article is an explanation of how it benefits semiparametric regression analysis. Even though we concentrate on semiparametric regression, the principles apply quite generally and can be transferred to other classes of statistical models such as those involving, for example, missing data, time series correlation structures and classification-oriented loss functions.

The efficiencies afforded by VMP also apply to another message passing algorithm known as *expectation propagation* (e.g., Minka 2005), although here we focus on the simpler VMP approach. The high-quality software package Infer.NET (Minka et al., 2014) supports expectation propagation and VMP fitting of various Bayesian hierarchical models. However, the nature of MFVB/VMP is such that coverage of various arbitrary scenarios in a general purpose software package is virtually impossible. The current release of Infer.NET has limitations in that many important semiparametric regression scenarios are not supported and self-implementation is the only option. Therefore, it is important to understand the message passing paradigm and how it can be used to build both general purpose and special purpose approximate inference engines. This article is a launch pad for the algebra and computing required for fitting arbitrary semiparametric regression models, and other statistical models, regardless of support by Infer.NET. At first glance, the algebra of VMP is foreign-looking for readers who work in statistics. Section 3 provides the details of VMP for a Bayesian linear regression model and working through it carefully is recommended for digestion of the concept.

Recently, Kucukelbir et al. (2016) announced support of Gaussian variational approximations in the Stan package (Stan Development Team 2016). This is a different type of approximation used by Infer.NET and this article.

Mean field restrictions, upon which MFVB/VMP is based, often lead to much simpler approximate Bayesian inference algorithms compared with the unrestricted exact case. The accuracy of the inference is typically very good (e.g., Faes, Ormerod, and Wand 2011, Luts and Wand 2015). Nevertheless, mean field variational inference is prone to varying degrees of inaccuracy and, for classes of models of interest, benchmarking against Markov chain Monte Carlo fitting is recommended to see if the accuracy of MFVB/VMP is acceptable for the intended application. In Sections 4 and 5, we show how a wide variety of Gaussian, Bernoulli and Poisson response semiparametric models can be accommodated via a few updating rules. Moreover, the updates involve purely matrix algebraic manipulations and can be readily implemented, and compartmentalized into a small number of functions, in the analyst's computing environment of choice.

As explained in Section 3.5 of Winn and Bishop (2005), the messages required for VMP fitting can be passed according to a flexible schedule with convergence occurring, under mild conditions, regardless of the order in which the messages are updated. This entails straightforward parallelizability of VMP algorithms, meaning that for large models the computing can be distributed across several cores. Luts (2015) contained details on parallelization of variational semiparametric regression analysis for distributed datasets. In a similar vein, VMP can achieve real-time fitting and inference for semiparametric regression by analogy with the MFVB approaches described by Luts, Broderick and Wand (2014).

Section 2 provides background material relevant to VMP. In Section 3, we use a Bayesian linear regression setting to convey the main ideas of VMP and then describe the ease of extension to larger models. Sections 4 and 5 form the centerpiece of this article. They describe eight factor graph fragments that are the building blocks of a wide range of arbitrarily large semiparametric regression models. The more straightforward Gaussian response case is treated first in Section 4 and then, in Section 5, we show how Bernoulli and Poisson response models can also be accommodated via the addition of only a handful of algebraic rules. Speed considerations are briefly discussed in Section 6 before some concluding remarks in Section 7. An online supplement to this article provides technicalities such as detailed derivations.

2. Background Material

Here, we provide some notation and coverage of background material required for our treatment of VMP for semiparametric regression in upcoming sections.

2.1. Density Function Notation

In keeping with the MFVB and VMP literature, we let p be the generic symbol for a density function when describing models and exact posterior density functions. Approximate posterior density functions according to MFVB/VMP restrictions are denoted generically by q.

As an example, consider a model having observed data vector $\boldsymbol{y}$ and parameter vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. The joint posterior density function of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ is

$$p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\boldsymbol{y}) = \frac{p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{y})}{p(\boldsymbol{y})}.$$

A mean field approximation to $p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\boldsymbol{y})$, based on the restriction that $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ have posterior independence, is denoted by $q(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_2)$ with the dependence on $\boldsymbol{y}$ suppressed. The essence of mean field approximation and references to more detailed descriptions are given in Section 3.1.

2.2. Matrix Definitions and Results

If $\boldsymbol{v}$ is a column vector then $\|\boldsymbol{v}\| \equiv \sqrt{\boldsymbol{v}^T\boldsymbol{v}}$. For a $d \times d$ matrix $\boldsymbol{A}$ we let $\text{vec}(\boldsymbol{A})$ denote the $d^2 \times 1$ vector obtained by stacking the columns of $\boldsymbol{A}$ underneath each other in order from left to right. For a $d^2 \times 1$ vector $\boldsymbol{a}$ we let $\text{vec}^{-1}(\boldsymbol{a})$ denote the $d \times d$ matrix formed from listing the entries of $\boldsymbol{a}$ in a columnwise fashion in order from left to right. Note that vec^{-1} is the usual function inverse when the domain of vec is restricted to square matrices. In particular, $\text{vec}^{-1}\{\text{vec}(\boldsymbol{A})\} = \boldsymbol{A}$ for $d \times d$ matrices $\boldsymbol{A}$ and $\text{vec}\{\text{vec}^{-1}(\boldsymbol{a})\} = \boldsymbol{a}$ for $d^2 \times 1$ vectors $\boldsymbol{a}$. The following identity links vec and the matrix trace: $\text{tr}(\boldsymbol{A}^T\boldsymbol{B}) = \text{vec}(\boldsymbol{A})^T\text{vec}(\boldsymbol{B})$ for any two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ such that $\boldsymbol{A}^T\boldsymbol{B}$ is defined and square. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are both $d \times 1$ vectors then $\boldsymbol{a} \odot \boldsymbol{b}$ denotes their elementwise product and $\boldsymbol{a}/\boldsymbol{b}$ denotes their elementwise quotient. Lastly, we use the convention that function evaluation is elementwise when applied to vectors. For example, if $s : \mathbb{R} \to \mathbb{R}$ then $s(\boldsymbol{a})$ denotes the $d \times 1$ vector with ith entry equal to $s(a_i)$.

2.3. Exponential Family Distributions

Univariate exponential family density and probability mass functions are those that can be written in the form

$$p(x) = \exp\{\boldsymbol{T}(x)^T\boldsymbol{\eta} - A(\boldsymbol{\eta})\}h(x) \tag{1}$$

where $\boldsymbol{T}(x)$ is the *sufficient statistic*, $\boldsymbol{\eta}$ is the *natural parameter*, $A(\boldsymbol{\eta})$ is the *log-partition function,* and $h(x)$ is the *base measure*. Note that the sufficient statistic is not unique. However, it is common to take $\boldsymbol{T}(x)$ to be the simplest possible algebraic form given $p(x)$.

An exponential family density function that arises several times in this article is that corresponding to an *Inverse Chi-Squared* random variable. The density function has general form

$$p(x) = \{(\lambda/2)^{\kappa/2}/\Gamma(\kappa/2)\}\, x^{-(\kappa/2)-1} \exp\{-(\lambda/2)/x\}, \quad x > 0, \tag{2}$$

where $\kappa > 0$ and $\lambda > 0$ are, respectively, shape and scale parameters. Simple algebraic manipulations show that (2) is a special case of (1) with

$$\boldsymbol{T}(x) = \begin{bmatrix} \log(x) \\ 1/x \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}(\kappa+2) \\ -\frac{1}{2}\lambda \end{bmatrix} \quad \text{and} \quad h(x) = I(x > 0),$$

where $I(\mathcal{P}) = 1$ if $\mathcal{P}$ is true and $I(\mathcal{P}) = 0$ if $\mathcal{P}$ is false. The log-partition function is $A(\boldsymbol{\eta}) = (\eta_1 + 1)\log(-\eta_2) + \log\Gamma(-\eta_1 - 1)$.

Section S.1 of the online supplement chronicles the sufficient statistics and natural parameter vectors, and other relevant relationships, for several exponential family distributions arising in semiparametric regression. Included is extension to multivariate density functions for random vectors and matrices.

2.4. Factor Graphs

A *factor graph* is a graphical representation of the factor/argument dependencies of a real-valued function. Consider, for example, the function h defined on $\mathbb{R}^5$ as follows:

$$\begin{aligned} h(x_1, x_2, x_3, x_4, x_5) &\equiv (x_1 + x_2)\, \sin(x_2 + 3^{x_3 x_4}) \\ &\quad \times \sqrt{\frac{x_3}{x_3^3 + 1}}\, \tan^8(x_4^2 - x_5)\, \coth\left(\frac{x_5 + 9}{7x_1 + 1}\right) \\ &= f_1(x_1, x_2)\, f_2(x_2, x_3, x_4)\, f_3(x_3)\, f_4(x_4, x_5)\, f_5(x_1, x_5) \end{aligned} \tag{3}$$

where, for example, $f_1(x_1, x_2) \equiv x_1 + x_2$ and $f_2, \ldots, f_5$ are defined similarly. Then, Figure 1 shows a factor graph corresponding to h. The circular nodes match the arguments of h and the square nodes coincide with the factors in (3). Edges are drawn between each factor node and arguments of that factor. Factor graphs of functions are not unique since, for example, f_1 and f_2 could be combined into a single factor and a different factor graph would result.

All of the factor graphs in the remainder of this article are such that the circular nodes correspond to random variables, random vectors, and random matrices. Hence, we use the phrase

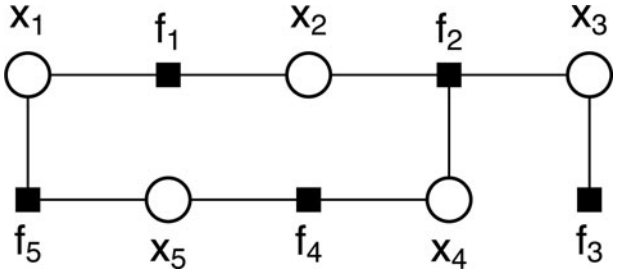


Figure 1. A factor graph corresponding to the function $h(x_1, x_2, x_3, x_4, x_5)$ defined by (3).

stochastic node to describe a circular node. A square node is simply called a *factor*. We use the word *node* to describe either a stochastic node or a factor. If two nodes on a factor graph are joined by an edge then we say that the nodes are *neighbors* of each other.

2.5. Variational Message Passing

Consider a Bayesian statistical model with observed data $\boldsymbol{D}$ and parameter vector $\boldsymbol{\theta}$. A mean field variational approximation to the posterior density function $p(\boldsymbol{\theta}|\boldsymbol{D})$ is

$$p(\boldsymbol{\theta}|\boldsymbol{D}) \approx q^*(\boldsymbol{\theta})$$

where $q^*(\boldsymbol{\theta})$ is the minimizer of the Kullback–Leibler divergence $\int q(\boldsymbol{\theta}) \log\{\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{D})}\}\, d\boldsymbol{\theta}$ subject to the product density restriction $q(\boldsymbol{\theta}) = \prod_{i=1}^{M} q(\boldsymbol{\theta}_i)$ and

$$\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\} \tag{4}$$

is some partition of $\boldsymbol{\theta}$. A useful notation for any subset S of $\{1, \ldots, M\}$ is $\boldsymbol{\theta}_S \equiv \{\boldsymbol{\theta}_i : i \in S\}$. Given the partition (4), the joint density function of $\boldsymbol{\theta}$ and $\boldsymbol{D}$ is expressible as

$$p(\boldsymbol{\theta}, \boldsymbol{D}) = \prod_{j=1}^{N} f_j(\boldsymbol{\theta}_{S_j}) \text{ for subsets } \mathrm{S_j} \text{ of } \{1, \ldots, \mathrm{M}\} \quad \text{and} \quad \text{factors } f_j,\ 1 \le j \le N. \tag{5}$$

For example, if $p(\boldsymbol{\theta}, \boldsymbol{D})$ is a directed acyclic graphical model with nodes $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M$ and $\boldsymbol{D}$ then

$$p(\boldsymbol{\theta}, \boldsymbol{D}) = \left\{\prod_{i=1}^{M} p(\boldsymbol{\theta}_i|\text{parents of } \boldsymbol{\theta}_i)\right\} p(\boldsymbol{D}|\text{ parents of } \boldsymbol{D}) \tag{6}$$

is an $N = M + 1$ example of (5) with f_j, $1 \le j \le M$, corresponding to the density function of $\boldsymbol{\theta}_j$ conditional on its parents and f_{M+1} corresponding to the likelihood. Each factor is a function of the subset of (4) corresponding to parental relationships in the directed acyclic graph. Further factorization of (6) may be possible.

The factor graph in Figure 2 shows an $M = 9, N = 11$ example of (5). The edges link each factor to the stochastic nodes on which the factor depends.

VMP can be expressed in terms of updating messages passed between nodes on the factor graph, and its description benefits from the notation:

$$\text{neighbors}(j) \equiv \{1 \le i \le M : \boldsymbol{\theta}_i \text{ is a neighbor of } f_j\}.$$

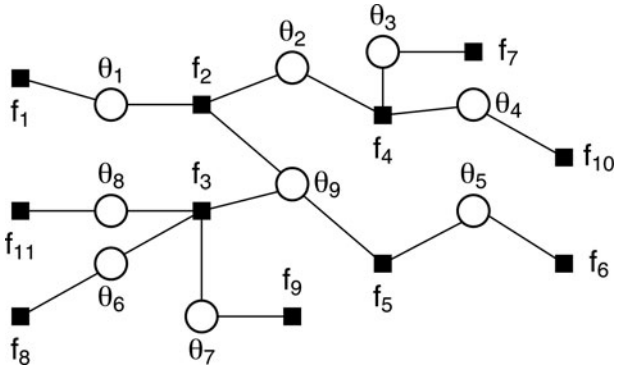


Figure 2. A factor graph corresponding to a Bayesian model with stochastic nodes $\theta_1, \ldots, \theta_9$ and factors $f_1, \ldots, f_{11}$.

Examples of this notation for the Figure 2 factor graph are

$$\text{neighbors}(1) = \{1\}, \quad \text{neighbors}(2) = \{1, 2, 9\} \quad \text{and}$$

$$\text{neighbors}(3) = \{6, 7, 8, 9\}.$$

Hence, according to this notation, $p(\boldsymbol{\theta}, \boldsymbol{D}) = \prod_{j=1}^{N} f_j(\boldsymbol{\theta}_{\text{neighbors}(j)})$. For each $1 \le i \le M$ and $1 \le j \le N$, the VMP stochastic node to factor message updates are

$$m_{\boldsymbol{\theta}_i \to f_j}(\boldsymbol{\theta}_i) \longleftarrow \propto \prod_{j' \neq j:\, i \in \text{neighbors}(j')} m_{f_{j'} \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \tag{7}$$

and the factor to stochastic node message updates are

$$m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \longleftarrow \propto \exp\left[E_{f_j \to \boldsymbol{\theta}_i}\left\{\log f_j(\boldsymbol{\theta}_{\text{neighbors}(j)})\right\}\right], \tag{8}$$

where $E_{f_j \to \boldsymbol{\theta}_i}$ denotes expectation with respect to the density function

$$\frac{\prod_{i' \in \text{neighbors}(j)\setminus\{i\}} m_{f_j \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'}) m_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'})}{\prod_{i' \in \text{neighbors}(j)\setminus\{i\}} \int m_{f_j \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'}) m_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'})\, d\boldsymbol{\theta}_{i'}}. \tag{9}$$

In (7) and (8) the $\longleftarrow \propto$ symbol means that the function of $\boldsymbol{\theta}_i$ on the left-hand side is updated according to the expression on the right-hand side but that multiplicative factors not depending on $\boldsymbol{\theta}_i$ can be ignored. For common statistical models, the messages arising from (8) are proportional to exponential family density functions and some simple examples are given in Section 3.2. If $\text{neighbors}(j)\setminus\{i\} = \emptyset$ then the expectation in (8) can be dispensed with and the right-hand side of (8) is proportional to $f_j(\boldsymbol{\theta}_{\text{neighbors}(j)})$. The normalizing factor in (9) involves summation if some of the $\boldsymbol{\theta}_{i'}$ have discrete components. Upon convergence of the messages, the Kullback–Leibler optimal q-densities are obtained via

$$q^*(\boldsymbol{\theta}_i) \propto \prod_{j: i \in \text{neighbors}(j)} m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i). \tag{10}$$

The genesis of (7)–(10) is given in Minka (2005) where a factor graph-based approach to VMP is described. Winn and Bishop (2005) developed an alternative version of VMP based on directed acyclic graphs. Yet another version of VMP is given in Appendix A of Minka and Winn (2008) which is similar, but not identical to, that given in Minka (2005). All three versions, as well as MFVB, converge to the same posterior density function approximations.

Section 3.6 of Winn and Bishop (2005) and Appendix A of Minka and Winn (2008) also describe the calculation of the marginal log-likelihood lower bound

$$\log \underline{p}(\boldsymbol{D}; q) \equiv \int q(\boldsymbol{\theta}) \log\left\{\frac{p(\boldsymbol{D}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})}\right\} d\boldsymbol{\theta} \tag{11}$$

which satisfies $\log \underline{p}(\boldsymbol{D}; q) \le \log\, p(\boldsymbol{D})$ regardless of q. In Winn and Bishop (2005), $\log \underline{p}(\boldsymbol{D}; q)$ is referred to as the *log evidence*. Section S.2.5 of the online supplement describes the streamlined computation of this quantity within the VMP framework.

2.6. Bayesian Semiparametric Regression

Detailed descriptions of Bayesian semiparametric regression are given in, for example, Chapter 16 of Ruppert, Wand, and Carroll (2003), Gurrin, Scurrah, and Hazelton (2005), and Wand (2009). Here, we provide a very brief account of the topic.

A fundamental ingredient, which facilitates the incorporation of nonlinear predictor effects, is that of *mixed model-based penalized splines*. If x is a continuous predictor variable then the most common form of a mixed model-based penalized spline in x is

$$f(x) = \beta_0 + \beta_1 x + \sum_{k=1}^{K} u_k z_k(x), \quad u_k | \sigma_u \overset{\text{ind.}}{\sim} N(0, \sigma_u^2), \quad 1 \le k \le K, \tag{12}$$

where $\{z_k : 1 \le k \le K\}$ is a suitable spline basis. A good default choice for the z_ks are canonical cubic O'Sullivan splines as described in Section 4 of Wand and Ormerod (2008), although any scatterplot smoother with a linear basis expansion and a single quadratic penalty can be reparametrized to have form (12).

In Bayesian semiparametric regression, β_0, β_1, and σ_u are random variables which require prior distributions to be imposed upon them. A common choice for (β_0, β_1) is a Bivariate Normal distribution prior, which allows straightforward approximate noninformativity to be imposed. As explained in Gelman (2006), approximate noninformativity of σ_u can be achieved via Uniform distribution and Half t distribution priors. The illustrations given in the current article use Half Cauchy priors for standard deviation parameters such as σ_u. This entails setting $p(\sigma_u) = 2/[\pi\, A_u\{1 + (\sigma_u/A_u)^2\}]$, $\sigma_u > 0$, where the scale parameter $A_u > 0$ is a user-specified hyperparameter. We denote this by $\sigma_u \sim \text{Half-Cauchy}(A)$. MFVB and VMP benefit from the following auxiliary variable result:

$$\begin{aligned} \text{if } \sigma_u^2 |\, a_u &\sim \text{Inverse-}\chi^2(1, 1/a_u) \quad \text{and} \\ a_u &\sim \text{Inverse-}\chi^2(1, 1/A_u^2) \\ \text{then} \quad \sigma_u &\sim \text{Half-Cauchy}(A_u). \end{aligned} \tag{13}$$

A covariance matrix extension of (13) is described in Huang and Wand (2013) and is given by (30) in Section 4.1.3.

The presence of penalized univariate or multivariate splines and, occasionally, penalized versions of other types of basis functions such as wavelets (e.g., Wand and Ormerod 2011) is the distinguishing feature of semiparametric regression compared with parametric regression. We advocate a broad view of the latter with linear models, linear mixed models, and their

various generalized response extensions included. According to this viewpoint, Bayesian versions of many of the models used in longitudinal and multilevel data analysis (e.g., Diggle et al., 2002; Fitzmaurice et al., 2008; Gelman and Hill 2007; Goldstein 2010) lie within the realm of Bayesian semiparametric regression.

2.7. A Central Function: G_{VMP}

For a $d \times 1$ vector $\boldsymbol{v}_1$ and a $d^2 \times 1$ vector $\boldsymbol{v}_2$ such that $\text{vec}^{-1}(\boldsymbol{v}_2)$ is symmetric, the following function is central to VMP for semiparametric regression:

$$G_{\text{VMP}}\left(\begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\end{bmatrix}; \boldsymbol{Q}, \boldsymbol{r}, s\right) \equiv -\frac{1}{8}\text{tr}\Big(\boldsymbol{Q}\{\text{vec}^{-1}(\boldsymbol{v}_2)\}^{-1}[\boldsymbol{v}_1\boldsymbol{v}_1^T\{\text{vec}^{-1}(\boldsymbol{v}_2)\}^{-1} - 2\boldsymbol{I}]\Big) - \frac{1}{2}\boldsymbol{r}^T\{\text{vec}^{-1}(\boldsymbol{v}_2)\}^{-1}\boldsymbol{v}_1 - \frac{1}{2}s. \quad (14)$$

The secondary arguments of G_{VMP} are a $d \times d$ matrix $\boldsymbol{Q}$, a $d \times 1$ vector $\boldsymbol{r}$ and $s \in \mathbb{R}$. The function G_{VMP} arises from the following fact: if $\boldsymbol{\theta}$ is a $d \times 1$ Multivariate Normal random vector with natural parameter vector $\boldsymbol{\eta}$ as defined by (S.4) of the online supplement then

$$E_{\boldsymbol{\theta}}\Big\{-\frac{1}{2}\big(\boldsymbol{\theta}^T\boldsymbol{Q}\boldsymbol{\theta} - 2\boldsymbol{r}^T\boldsymbol{\theta} + s\big)\Big\} = E_{\boldsymbol{\theta}}\Big(-\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{Q}\boldsymbol{\theta} + \boldsymbol{r}^T\boldsymbol{\theta}\Big) - \frac{1}{2}s = G_{\text{VMP}}(\boldsymbol{\eta}; \boldsymbol{Q}, \boldsymbol{r}, s).$$

For example, if $\boldsymbol{a}$ is an $m \times 1$ vector and $\boldsymbol{A}$ is an $m \times d$ matrix then

$$E_{\boldsymbol{\theta}}\left\{-\frac{1}{2}\|\boldsymbol{a} - \boldsymbol{A}\boldsymbol{\theta}\|^2\right\} = G_{\text{VMP}}\Big(\boldsymbol{\eta}; \boldsymbol{A}^T\boldsymbol{A},\ \boldsymbol{A}^T\boldsymbol{a},\ \boldsymbol{a}^T\boldsymbol{a}\Big).$$

3. Linear Regression Illustrative Example

Consider the Bayesian regression model

$$\boldsymbol{y}|\,\boldsymbol{\beta}, \sigma^2 \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}), \quad \boldsymbol{\beta} \sim N(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}),$$
$$\sigma^2|a \sim \text{Inverse-}\chi^2(1, 1/a), \quad a \sim \text{Inverse-}\chi^2(1, 1/A^2) \quad (15)$$

where $\boldsymbol{y}$ is an $n \times 1$ vector of response data and $\boldsymbol{X}$ is an $n \times d$ design matrix. The $d \times 1$ vector $\boldsymbol{\mu}_{\boldsymbol{\beta}}$, the $d \times d$ covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ and $A > 0$ are user-specified hyperparameters that remain fixed throughout any approximate Bayesian inference procedure for (15). As explained in Section 2.6, the marginal prior distribution on σ in (15) is Half-Cauchy(A). The joint posterior density function of the model parameters and auxiliary variable a is

$$p(\boldsymbol{\beta}, \sigma^2, a\,|\,\boldsymbol{y}) = \frac{p(\boldsymbol{\beta}, \sigma^2, a, \boldsymbol{y})}{p(\boldsymbol{y})} \quad (16)$$

but is analytically intractable and numerically challenging. MCMC (e.g., Chapters 11-12, Gelman et al., 2014) is the most common tool for making approximate Bayesian inference for $\boldsymbol{\beta}$ and σ^2. The computationally intensive nature of MCMC entails that, while its speed will be acceptable for some applications, there are others where faster approximations are desirable or necessary. We next describe MFVB as one such fast alternative.

3.1. Mean Field Variational Bayes Approach

MFVB is a prescription for approximation of posterior density functions in a graphical model. References on MFVB for general graphical models include Bishop (2006), Wainwright and Jordan (2008) and Ormerod and Wand (2010). In this section, we focus on MFVB for approximation of (16). This is founded upon $p(\boldsymbol{\beta}, \sigma^2, a\,|\,\boldsymbol{y})$ being restricted to have the product form

$$q(\boldsymbol{\beta})\,q(\sigma^2)\,q(a) \quad (17)$$

for density functions $q(\boldsymbol{\beta})$, $q(\sigma^2)$, and $q(a)$. These *q-density functions* are then chosen to minimize the Kullback–Leibler distance between $p(\boldsymbol{\beta}, \sigma^2, a|\boldsymbol{y})$ and $q(\boldsymbol{\beta})\,q(\sigma^2)\,q(a)$:

$$\int q(\boldsymbol{\beta})\,q(\sigma^2)\,q(a)\,\log\left\{\frac{q(\boldsymbol{\beta})\,q(\sigma^2)\,q(a)}{p(\boldsymbol{\beta}, a, \sigma^2|\boldsymbol{y})}\right\} d\boldsymbol{\beta}\,d\sigma^2\,da.$$

One can then prove by variational calculus that the optimal q-densities satisfy:

$q^*(\boldsymbol{\beta})$ is a $N(\boldsymbol{\mu}_{q(\boldsymbol{\beta})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta})})$ density function,

$q^*(\sigma^2)$ is an Inverse-$\chi^2(n+1, \lambda_{q(\sigma^2)})$ density function, and

$q^*(a)$ is an Inverse-$\chi^2(2, \lambda_{q(a)})$ density function

for some $d \times 1$ vector $\boldsymbol{\mu}_{q(\boldsymbol{\beta})}$, $d \times d$ covariance matrix $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}$ and positive scalars $\lambda_{q(\sigma^2)}$ and $\lambda_{q(a)}$. These q-density parameters do not have closed form solutions but, instead, can be determined iteratively via coordinate ascent as explained in Section 10.1.1 of Bishop (2006) and Section 2.2 of Ormerod and Wand (2010). For the model at hand, the coordinate ascent updates reduce to Algorithm 1. Here and elsewhere, "⟵" indicates that the quantity on the left-hand side is updated according to the expression on the right-hand side.

3.2. Alternative Approach Based on Variational Message Passing

We now explain the VMP alternative for the Bayesian linear regression example. First, note that the joint distribution of all random variables in model (15) admits the factorization

$$p(\boldsymbol{y}, \boldsymbol{\beta}, \sigma^2, a) = p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta})p(\sigma^2|a)p(a). \quad (18)$$

Algorithm 1 *Mean field variational Bayes algorithm for approximate inference in the Gaussian response linear regression model (15).*

Initialize: $\lambda_{q(\sigma^2)} > 0$.
Cycle:

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})} \longleftarrow \left\{\left(\frac{n+1}{\lambda_{q(\sigma^2)}}\right)\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\right\}^{-1}$$

$$\boldsymbol{\mu}_{q(\boldsymbol{\beta})} \longleftarrow \left(\frac{n+1}{\lambda_{q(\sigma^2)}}\right)\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}\left(\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}\right)$$

$$\lambda_{q(a)} \longleftarrow 2\left\{\left(\frac{n+1}{\lambda_{q(\sigma^2)}}\right) + A^{-2}\right\}$$

$$\lambda_{q(\sigma^2)} \longleftarrow \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\mu}_{q(\boldsymbol{\beta})}^T\boldsymbol{X}^T\boldsymbol{y} + \text{tr}[(\boldsymbol{X}^T\boldsymbol{X})\{\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})} + \boldsymbol{\mu}_{q(\boldsymbol{\beta})}\boldsymbol{\mu}_{q(\boldsymbol{\beta})}^T\}] + 2/\lambda_{q(a)}$$

until the changes in all q-density parameters are negligible.

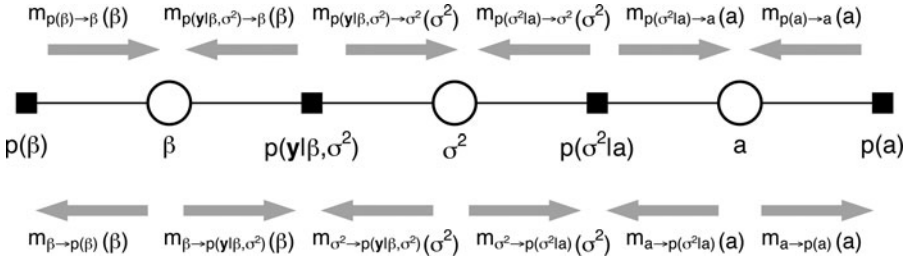


Figure 3. A factor graph for the function $p(y, \boldsymbol{\beta}, \sigma^2, a)$ with stochastic nodes $\boldsymbol{\beta}, \sigma^2$ and a, corresponding to the factors in product restriction (17). Also shown are each of the messages between neighboring nodes on the factor graph. The gray arrows depict the directions in which the messages are passed.

Treating (18) as a function of parameters corresponding to each factor in the mean field restriction (17) we arrive at the factor graph shown in Figure 3.

VMP iteration for fitting model (15) involves the updating of *messages* passed from each node in Figure 3 factor graph to its neighboring nodes. Each message is a function of the stochastic node that receives or sends the message. For example, the nodes σ^2 and $p(\sigma^2|a)$ are neighbors of each other in Figure 3 factor graph. The messages passed between these two nodes are both functions of the stochastic node σ^2 and are denoted by $m_{\sigma^2\to p(\sigma^2|a)}(\sigma^2)$ and $m_{p(\sigma^2|a)\to\sigma^2}(\sigma^2)$. The subscripts of m designates the nodes involved in the message passing and the direction in which the message is passed. Figure 3 shows all 12 of the messages between neighboring nodes on the factor graph.

Based on the VMP updating equations given in Section 2.5, and with details given in Section S.2.1 of the online supplement, the factor to stochastic node messages have the following functional forms after the first iteration:

$$
\begin{aligned}
m_{p(\boldsymbol{\beta})\to\boldsymbol{\beta}}(\boldsymbol{\beta}) &= \exp\left\{\begin{bmatrix}\boldsymbol{\beta}\\ \text{vec}(\boldsymbol{\beta}\boldsymbol{\beta}^T)\end{bmatrix}^T \boldsymbol{\eta}_{p(\boldsymbol{\beta})\to\boldsymbol{\beta}}\right\},\\
m_{p(y|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}(\boldsymbol{\beta}) &= \exp\left\{\begin{bmatrix}\boldsymbol{\beta}\\ \text{vec}(\boldsymbol{\beta}\boldsymbol{\beta}^T)\end{bmatrix}^T \boldsymbol{\eta}_{p(y|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}\right\},\\
m_{p(y|\boldsymbol{\beta},\sigma^2)\to\sigma^2}(\sigma^2) &= \exp\left\{\begin{bmatrix}\log(\sigma^2)\\ 1/\sigma^2\end{bmatrix}^T \boldsymbol{\eta}_{p(y|\boldsymbol{\beta},\sigma^2)\to\sigma^2}\right\},\\
m_{p(\sigma^2|a)\to\sigma^2}(\sigma^2) &= \exp\left\{\begin{bmatrix}\log(\sigma^2)\\ 1/\sigma^2\end{bmatrix}^T \boldsymbol{\eta}_{p(\sigma^2|a)\to\sigma^2}\right\},\\
m_{p(\sigma^2|a)\to a}(a) &= \exp\left\{\begin{bmatrix}\log(a)\\ 1/a\end{bmatrix}^T \boldsymbol{\eta}_{p(\sigma^2|a)\to a}\right\}\\
\text{and}\quad m_{p(a)\to a}(a) &= \exp\left\{\begin{bmatrix}\log(a)\\ 1/a\end{bmatrix}^T \boldsymbol{\eta}_{p(a)\to a}\right\}
\end{aligned}
\tag{19}
$$

for $(d + d^2) \times 1$ vectors $\boldsymbol{\eta}_{p(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}_{p(y|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ and 2×1 vectors $\boldsymbol{\eta}_{p(y|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$, $\boldsymbol{\eta}_{p(\sigma^2|a)\to\sigma^2}$, $\boldsymbol{\eta}_{p(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}_{p(a)\to a}$. The fixed form of the messages means that, for the remaining iterations, the message updates (7) and (8) simply involve updates for the natural parameter vectors of the messages. Note that the last four of these messages are proportional to Inverse Chi-Squared density functions. The first two are proportional to d-dimensional multivariate normal distributions, but expressed in exponential family form as explained in Section S.1.6 of the online supplement. Therefore, normalizing factors aside, each of the subscripted $\boldsymbol{\eta}$ vectors are natural parameters for a particular exponential family density function. The stochastic node to factor messages have the same functional forms as their reverse messages. For example,

$$
m_{\sigma^2\to p(\sigma^2|a)}(\sigma^2) = \exp\left\{\begin{bmatrix}\log(\sigma^2)\\ 1/\sigma^2\end{bmatrix}^T \boldsymbol{\eta}_{\sigma^2\to p(\sigma^2|a)}\right\}
$$

for some 2×1 vector $\boldsymbol{\eta}_{\sigma^2\to p(\sigma^2|a)}$.

Once the functional forms of the messages have been determined, the VMP iteration loop has the following generic steps:

1. Choose a factor.
2. Update the parameter vectors of the messages passed from the factor's neighboring stochastic nodes to the factor.
3. Update the parameter vectors of the messages passed from the factor to its neighboring stochastic nodes.

For typical semiparametric regression models the order in which factors are chosen does not matter although all factors should eventually be chosen as the iterations proceed. There are some classes of models, outside those treated in this article, for which local optima exist and the update order may affect which optimum is attained.

The updates of the stochastic node to factor natural parameter vectors have simple forms based on (7) and are updated as follows:

$$
\begin{aligned}
\boldsymbol{\eta}_{\boldsymbol{\beta}\to p(\boldsymbol{\beta})} &\longleftarrow \boldsymbol{\eta}_{p(y|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}},\\
\boldsymbol{\eta}_{\boldsymbol{\beta}\to p(y|\boldsymbol{\beta},\sigma^2)} &\longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{\beta})\to\boldsymbol{\beta}}\\
\boldsymbol{\eta}_{\sigma^2\to p(y|\boldsymbol{\beta},\sigma^2)} &\longleftarrow \boldsymbol{\eta}_{p(\sigma^2|a)\to\sigma^2},\\
\boldsymbol{\eta}_{\sigma^2\to p(\sigma^2|a)} &\longleftarrow \boldsymbol{\eta}_{p(y|\boldsymbol{\beta},\sigma^2)\to\sigma^2},\\
\boldsymbol{\eta}_{a\to p(\sigma^2|a)} &\longleftarrow \boldsymbol{\eta}_{p(a)\to a},\quad \text{and}\\
\boldsymbol{\eta}_{a\to p(a)} &\longleftarrow \boldsymbol{\eta}_{p(\sigma^2|a)\to a}.
\end{aligned}
\tag{20}
$$

Based on (8) and (9) $m_{p(\boldsymbol{\beta})\to\boldsymbol{\beta}}(\boldsymbol{\beta}) \propto p(\boldsymbol{\beta})$ and $m_{p(a)\to a}(a) \propto p(a)$ so the natural parameter updates for these two messages

are simply

$$\boldsymbol{\eta}_{p(\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ -\frac{1}{2}\text{vec}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}) \end{bmatrix} \quad \text{and}$$

$$\boldsymbol{\eta}_{p(a)\to a} \to \begin{bmatrix} -3/2 \\ -1/A^2 \end{bmatrix} \tag{21}$$

and remain constant throughout the iterations. The updates corresponding to the messages sent from $p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)$ to its neighboring stochastic nodes can be obtained from (8) and (9). The expectation in (8) reduces to a linear combination of expected sufficient statistics. Table S.1 of the online supplement gives the required expressions. Simple algebra then leads to

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}} \longleftarrow \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{y} \\ -\frac{1}{2}\text{vec}(\boldsymbol{X}^T\boldsymbol{X}) \end{bmatrix} \times \frac{\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}\right)_1 + 1}{\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}\right)_2}$$

$$\text{and} \quad \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} \longleftarrow \begin{bmatrix} -\frac{1}{2}\,n \\ G_{\text{VMP}}\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}}; \boldsymbol{X}^T\boldsymbol{X}, \boldsymbol{X}^T\boldsymbol{y}, \boldsymbol{y}^T\boldsymbol{y}\right) \end{bmatrix} \tag{22}$$

where

$$\begin{aligned} \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} &\equiv \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} + \boldsymbol{\eta}_{\sigma^2\to p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)}, \\ \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} &\equiv \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}} + \boldsymbol{\eta}_{\boldsymbol{\beta}\to p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)}, \end{aligned} \tag{23}$$

$(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2})_i$ denotes the ith entry of $\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}$ and G_{VMP} is explained in Section 2.7. The parameter updates for the messages passed from $p(\sigma^2|a)$ to its neighbors are

$$\boldsymbol{\eta}_{p(\sigma^2|a)\to\sigma^2} \longleftarrow -\frac{1}{2}\begin{bmatrix} 3 \\ \frac{\left(\boldsymbol{\eta}_{p(\sigma^2|a)\leftrightarrow a}\right)_1 + 1}{\left(\boldsymbol{\eta}_{p(\sigma^2|a)\leftrightarrow a}\right)_2} \end{bmatrix}$$

$$\text{and} \quad \boldsymbol{\eta}_{p(\sigma^2|a)\to a} \longleftarrow -\frac{1}{2}\begin{bmatrix} 1 \\ \frac{\left(\boldsymbol{\eta}_{p(\sigma^2|a)\leftrightarrow\sigma^2}\right)_1 + 1}{\left(\boldsymbol{\eta}_{p(\sigma^2|a)\leftrightarrow\sigma^2}\right)_2} \end{bmatrix}. \tag{24}$$

where the definitions of $\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}$ and $\boldsymbol{\eta}_{p(\sigma^2|a)\leftrightarrow a}$ are analogous to those given in (23).

After initializing the stochastic node to factor natural parameters, updates (20), (21), (22) and (24) form an iterative scheme in the message natural parameter space. Once convergence of the messages has been attained, the q-density natural parameters can be obtained from (10) as:

$$\begin{aligned} \boldsymbol{\eta}_{q(\boldsymbol{\beta})} &\longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{\beta})\to\boldsymbol{\beta}} + \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}, \\ \boldsymbol{\eta}_{q(\sigma^2)} &\longleftarrow \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} + \boldsymbol{\eta}_{p(\sigma^2|a)\to\sigma^2} \end{aligned}$$

$$\text{and} \quad \boldsymbol{\eta}_{q(a)} \longleftarrow \boldsymbol{\eta}_{p(\sigma^2|a)\to a} + \boldsymbol{\eta}_{p(a)\to a}. \tag{25}$$

Updates (25) show that the natural parameters of a q-density of a stochastic node and incoming messages to that node have simple linear relationships. This, together with (7), motivates working with natural parameters in VMP.

The q-density common parameters can be obtained from (25) using (S.3) and (S.4) of the online supplement and lead to

$$\begin{aligned} \boldsymbol{\mu}_{q(\boldsymbol{\beta})} &= -\frac{1}{2}\{\text{vec}^{-1}\left((\boldsymbol{\eta}_{q(\boldsymbol{\beta})})_2\right)\}^{-1}(\boldsymbol{\eta}_{q(\boldsymbol{\beta})})_1, \\ \boldsymbol{\Sigma}_{q(\boldsymbol{\beta})} &= -\frac{1}{2}\{\text{vec}^{-1}\left((\boldsymbol{\eta}_{q(\boldsymbol{\beta})})_2\right)\}^{-1}, \\ \lambda_{q(\sigma^2)} &= -2\left(\boldsymbol{\eta}_{q(\sigma^2)}\right)_2 \quad \text{and} \quad \lambda_{q(a)} = -2\left(\boldsymbol{\eta}_{q(a)}\right)_2, \end{aligned}$$

where $(\boldsymbol{\eta}_{q(\boldsymbol{\beta})})_1$ contains the first d entries of $\boldsymbol{\eta}_{q(\boldsymbol{\beta})}$ and $(\boldsymbol{\eta}_{q(\boldsymbol{\beta})})_2$ contains the remaining d^2 entries of $\boldsymbol{\eta}_{q(\boldsymbol{\beta})}$. The values of $\boldsymbol{\mu}_{q(\boldsymbol{\beta})}$, $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}$, $\lambda_{q(\sigma^2)}$ and $\lambda_{q(a)}$ are the same regardless of whether one uses the MFVB approach encapsulated in Algorithm 1 or the VMP approach described in this section. On face value, it would appear that the MFVB approach is superior due to its succinctness. However, this ranking of MFVB over VMP is within the confines of approximate inference for model (15). As we now explain in Section 3.2.1, VMP is a more attractive proposition when semiparametric regression models are extended arbitrarily.

3.2.1. Arbitrarily Large Model Viewpoint

We now turn attention to variational inference for arbitrarily large semiparametric regression models and how the message passing approach allows streamlining of the required calculations.

Figure 4 shows both simple linear regression and nonparametric regression fits to data on 93 passenger car models on sale in USA in 1993 (source: Lock 1993). The ith response observation (y_i) is fuel efficiency on city roads (miles/gallon) and the ith predictor observation (x_i) is weight of the car (pounds).

The simple linear regression fit is obtained using VMP applied to the special case of (15) with $\boldsymbol{X} = [1\ x_i]_{1\le i\le n}$. The nonparametric regression fit in Figure 4 is according to mixed model-based penalized spline model

$$\begin{aligned} \boldsymbol{y}|\,\boldsymbol{\beta},\boldsymbol{u},\sigma_\varepsilon^2 &\sim N(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{Z}\boldsymbol{u},\sigma_\varepsilon^2\boldsymbol{I}), \quad \boldsymbol{u}|\,\sigma_u^2 \sim N(0,\sigma_u^2\boldsymbol{I}), \\ \boldsymbol{\beta} &\sim N(\boldsymbol{\mu}_{\boldsymbol{\beta}},\boldsymbol{\Sigma}_{\boldsymbol{\beta}}), \\ \sigma_u^2|a_u &\sim \text{Inverse-}\chi^2(1,1/a_u), \quad a_u \sim \text{Inverse-}\chi^2(1,1/A_u^2), \\ \sigma_\varepsilon^2|\,a_\varepsilon &\sim \text{Inverse-}\chi^2(1,1/a_\varepsilon), \quad a_\varepsilon \sim \text{Inverse-}\chi^2(1,1/A_\varepsilon^2), \end{aligned} \tag{26}$$

where

$$\boldsymbol{Z} \equiv \begin{bmatrix} z_1(x_1) & \cdots & z_K(x_1) \\ \vdots & \ddots & \vdots \\ z_1(x_n) & \cdots & z_K(x_n) \end{bmatrix}$$

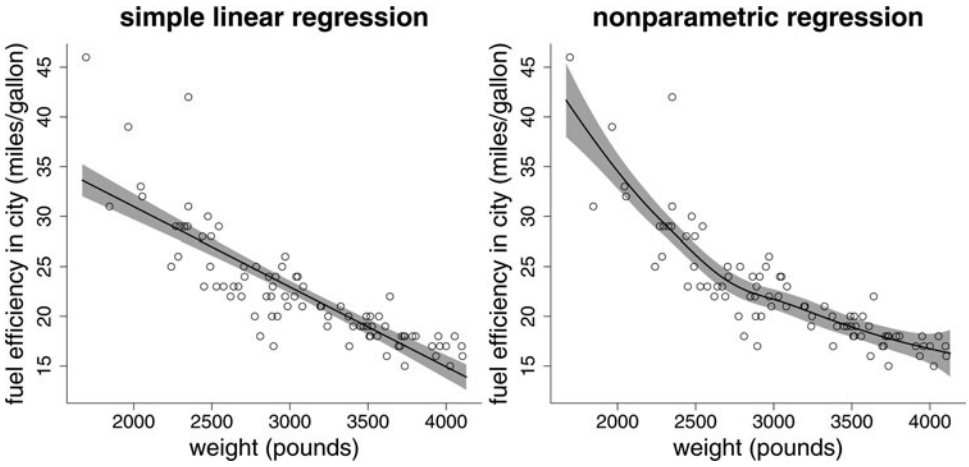


Figure 4. Left panel: VMP-based simple linear regression fit to data on fuel efficiency and weight of 93 passenger car models on sale in USA in 1993 (source: Lock 1993). The fitted line is the posterior mean and the shaded region shows pointwise 95% credible sets according to the mean field approximation (17). Right panel: similar to the left panel but for nonparametric regression according to the mixed model-based penalized spline extension (26) with mean field approximation (28).

for a spline basis $\{z_k : 1 \le k \le K\}$ as defined adjacent to (12). The mean field approximation being used here is

$$p(\boldsymbol{\beta}, \boldsymbol{u}, \sigma_u^2, a_u, \sigma_\varepsilon^2, a_\varepsilon|\boldsymbol{y}) \approx q(\boldsymbol{\beta}, \boldsymbol{u}, a_u, a_\varepsilon)q(\sigma_u^2, \sigma_\varepsilon^2). \quad (27)$$

However, further product density forms arise due to conditional independencies in the model (e.g., Section 10.2.5 of Bishop 2006) and it can be established that (27) is equivalent to

$$p(\boldsymbol{\beta}, \boldsymbol{u}, \sigma_u^2, a_u, \sigma_\varepsilon^2, a_\varepsilon|\boldsymbol{y}) \approx q(\boldsymbol{\beta}, \boldsymbol{u})q(\sigma_u^2)q(a_u)\, q(\sigma_\varepsilon^2)q(a_\varepsilon). \quad (28)$$

The extension of the VMP updates when transitioning from the linear regression model (15) to (26) benefits from:

Definition. A *factor graph fragment*, or *fragment* for short, is a subgraph of a factor graph consisting of a single factor and each of the stochastic nodes that are neighbors of the factor.

Figure 5 shows the factor graph corresponding to (26) with mean field approximation (28). This factor graph has six factors and therefore six fragments. Five of them have the same form as the fragments of the factors of Figure 3 and are colored gray. The black-colored fragment corresponds to the following penalization of the coefficient vector:

$$\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{u} \end{bmatrix} \Big| \sigma_u^2 \sim N\left(\begin{bmatrix} \sigma_\beta^2 \boldsymbol{I}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_u^2 \boldsymbol{I}_K \end{bmatrix}\right)$$

and is a distributional form that does not appear in the linear regression model.

The stochastic node to factor messages in Figure 5 have trivial updates analogous to those given in (20). The factor to stochastic messages are more complicated, but for the five fragments shown in gray in Figure 5 they are identical or very similar to analogous updates in Figure 3 factor graph, as we now explain:

1. The message passed from $p(a_u)$ to a_u has the same form as that passed from $p(a)$ to a for model (15). The natural parameter updates $\boldsymbol{\eta}_{p(a_u)\to a_u}$ takes the same form as that for $\boldsymbol{\eta}_{p(a)\to a}$ in (21).
2. Comments similar to those given in (1) apply to the messages passed from $p(a_\varepsilon)$ to a_ε.
3. The messages passed from the factor $p(\sigma_u^2|\, a_u)$ to its neighboring stochastic nodes σ_u^2 and a_u have the same form as those passed from $p(\sigma^2|a)$ to σ^2 and a for model (15). The natural parameter updates $\boldsymbol{\eta}_{p(\sigma_u^2|a_u)\to\sigma_u^2}$ and $\boldsymbol{\eta}_{p(\sigma_u^2|\, a_u)\to a_u}$ take the same forms as those for $\boldsymbol{\eta}_{p(\sigma^2|\, a)\to\sigma^2}$ and $\boldsymbol{\eta}_{p(\sigma^2|\, a)\to a}$ in (24).
4. Comments similar to those given in (1) apply to the messages passed from $p(\sigma_\varepsilon^2|a_\varepsilon)$ to its neighboring stochastic nodes.
5. The messages passed from the factor $p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{u}, \sigma_\epsilon^2)$ to its neighboring stochastic nodes $(\boldsymbol{\beta}, \boldsymbol{u})$ and σ_ϵ^2 have a similar form to those passed from $p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)$ to β and σ^2 for model (15). The natural parameter updates $\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$ take the same forms as those for $\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{u},\sigma_\varepsilon^2)\to(\boldsymbol{\beta},\boldsymbol{u})}$ and $\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\beta},\boldsymbol{u},\sigma_\varepsilon^2)\to\sigma_\varepsilon^2}$ in (22)

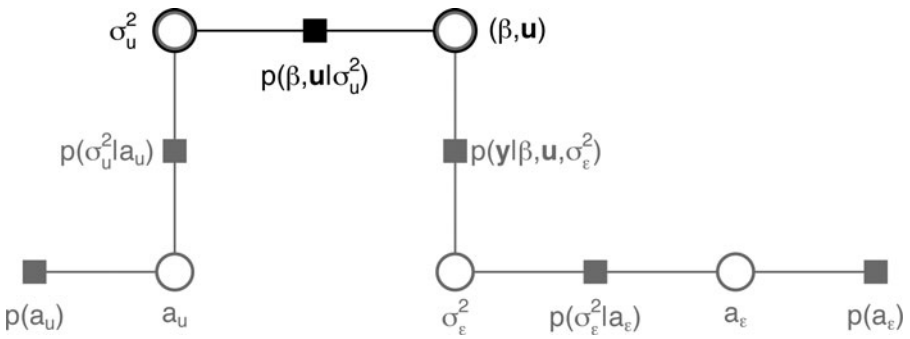


Figure 5. Diagrammatic depiction of the extension from simple linear regression to penalized spline regression. The fragment shown in black is the only one that is of a different type compared with the fragments in the Bayesian linear regression model. The fragments shown in gray are present in the linear model factor graph shown in Figure 3.

but with $\boldsymbol{\beta}$ replaced by $(\boldsymbol{\beta}, \boldsymbol{u})$, σ^2 replaced by σ_ε^2 and $\boldsymbol{X}$ replaced by $\boldsymbol{C} \equiv [\boldsymbol{X}\ \boldsymbol{Z}]$.

It remains to take care of the black-colored fragment of Figure 5. The message passed from $p(\boldsymbol{\beta}, \boldsymbol{u}|\sigma_u^2)$ to σ_u^2 is

$$m_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\to\sigma_u^2}(\sigma_u^2) = \exp\left\{\begin{bmatrix}\log(\sigma_u^2)\\ 1/\sigma_u^2\end{bmatrix}^T \boldsymbol{\eta}_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\to\sigma_u^2}\right\},$$

where

$$\boldsymbol{\eta}_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\to\sigma_u^2} \longleftarrow \begin{bmatrix}-K/2\\ G_{\text{VMP}}\left(\boldsymbol{\eta}_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\leftrightarrow(\boldsymbol{\beta},\boldsymbol{u})}; \boldsymbol{D}, \boldsymbol{0}, 0\right)\end{bmatrix},$$

where $\boldsymbol{D} \equiv \text{diag}(\boldsymbol{0}_2, \boldsymbol{1}_K)$ and the function G_{VMP} is defined by (14). The message passed from $p(\boldsymbol{\beta}, \boldsymbol{u}|\sigma_u^2)$ to $(\boldsymbol{\beta}, \boldsymbol{u})$ will be shown (Section 4.1.4) to equal

$$m_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\to(\boldsymbol{\beta},\boldsymbol{u})}(\boldsymbol{\beta}, \boldsymbol{u}) = \exp\left\{\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{u}\\ \text{vec}\left(\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{u}\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{u}\end{bmatrix}^T\right)\end{bmatrix}^T \boldsymbol{\eta}_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\to(\boldsymbol{\beta},\boldsymbol{u})}\right\}$$

with natural parameter update

$$\boldsymbol{\eta}_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\to(\boldsymbol{\beta},\boldsymbol{u})} \longleftarrow \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}\\ \boldsymbol{0}_K\\ -\frac{1}{2}\text{vec}\left(\text{blockdiag}\left(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}, \left\{\frac{\left(\boldsymbol{\eta}_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\leftrightarrow\sigma_u^2}\right)_1 + 1}{\left(\boldsymbol{\eta}_{p(\boldsymbol{\beta},\boldsymbol{u}|\,\sigma_u^2)\leftrightarrow\sigma_u^2}\right)_2}\right\}\boldsymbol{I}_K\right)\right)\end{bmatrix}.$$

In Section 4, we catalog fragment types and identify five that are fundamental to semiparametric regression analysis via MFVB/VMP. The form of the factor to stochastic node updates for these fragments only needs to be derived and implemented once if developing a suite of programs for VMP-based semiparametric regression. Such cataloging allows for arbitrarily large models to be handled without an onerous algebraic and computational overhead.

3.2.2. Conjugate Factor Graphs

We will say that a factor graph corresponding to a variational message passing scheme is *conjugate* if, for each stochastic node, the messages passed to the node are in the same exponential family. The two factor graphs of this section, shown in Figures 3 and 5, are conjugate factor graphs. For example, it is apparent from (19) that, in Figure 3, the two messages passed to σ^2 are both proportional to Inverse Chi-Squared density functions. However, some of the exponential forms do not correspond to proper density functions. In Figure 3, the convergent form of $m_{p(\sigma^2|a)\to a}(a)$ is

$$m_{p(\sigma^2|a)\to a}(a) = \exp\left\{\begin{bmatrix}\log(a)\\ 1/a\end{bmatrix}^T \begin{bmatrix}-\frac{1}{2}\\ -\lambda_{p(\sigma^2|\,a)\to a}\end{bmatrix}\right\} \quad \text{for some} \quad \lambda_{p(\sigma^2|\,a)\to a} > 0$$

which is not proportional to a proper density function.

The concept of a conjugate factor graph can be extended to subgraphs of the factor graph at hand, in that conjugacy holds in some parts of a factor graph but not necessarily in other parts.

4. Gaussian Response Semiparametric Regression

Since many popular Gaussian response semiparametric regression models admit conjugate factor graphs, we first focus on their fitting via VMP. Generalized response models are more challenging and their treatment is postponed until Section 5. We start by identifying five fundamental fragments.

4.1. Five Fundamental Fragments

Table 1 shows five factor graph fragments that are fundamental to VMP-based semiparametric regression. We use generic notation, such as $\boldsymbol{\theta}$ for a random vector and $\boldsymbol{A}$ for a design matrix, rather than notation that matches specific semiparametric regression models. This is in keeping with update formulas within fragments being the building blocks for the handling of arbitrarily large models.

4.1.1. Gaussian Prior Fragment

The *Gaussian prior fragment* corresponds to the following prior specification of the $d^\theta \times 1$ random vector $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} \sim N(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}).$$

The $d^\theta \times 1$ vector $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and $d^\theta \times d^\theta$ covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ are user-specified hyperparameters. The fragment is shown in Table 1 and has factor

$$p(\boldsymbol{\theta}) = (2\pi)^{-d^\theta/2}|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu}_{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_{\boldsymbol{\theta}})\right\}$$

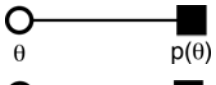
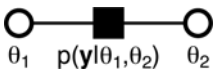
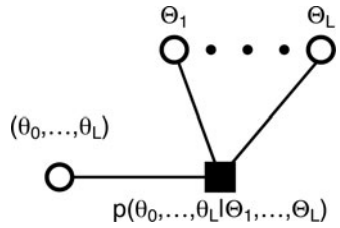
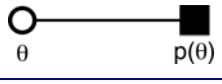
and the single stochastic node $\boldsymbol{\theta}$. Using the natural form of the Multivariate Normal distribution described in Section S.1.6 of the online supplement, the factor to stochastic node message is proportional to $p(\boldsymbol{\theta})$ and has natural parameter form:

$$m_{p(\boldsymbol{\theta})\to\boldsymbol{\theta}}(\boldsymbol{\theta}) = \exp\left\{\begin{bmatrix}\boldsymbol{\theta}\\ \text{vec}(\boldsymbol{\theta}\boldsymbol{\theta}^T)\end{bmatrix}^T \boldsymbol{\eta}_{p(\boldsymbol{\theta})\to\boldsymbol{\theta}}\right\}.$$

The natural parameter vector is a fixed vector depending only on the hyperparameters:

$$\boldsymbol{\eta}_{p(\boldsymbol{\theta})\to\boldsymbol{\theta}} \longleftarrow \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\theta}}\\ -\frac{1}{2}\text{vec}(\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1})\end{bmatrix}.$$

Table 1. Five fundamental factor graph fragments for Gaussian response semiparametric regression.

Fragment name	Diagram	Distributional statement
1. Gaussian prior		$\boldsymbol{\theta} \sim N(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$
2. Inverse Wishart prior		$\boldsymbol{\Theta} \sim$ Inverse-Wishart$(\kappa_{\boldsymbol{\Theta}}, \boldsymbol{\Lambda}_{\boldsymbol{\Theta}})$
3. Iterated Inverse G-Wishart		$\boldsymbol{\Theta}_1 \vert \boldsymbol{\Theta}_2 \sim$ Inverse-G-Wishart$(G, \kappa, \boldsymbol{\Theta}_2^{-1})$
4. Gaussian penalization		$\begin{bmatrix} \boldsymbol{\theta}_0 \\ \vdots \\ \boldsymbol{\theta}_L \end{bmatrix} \Big\vert \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_L \sim N\left(\begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{\theta}_0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\theta}_0} & \boldsymbol{O}^T \\ \boldsymbol{O} & \underset{1 \le \ell \le L}{\text{blockdiag}} (\boldsymbol{I}_{m_\ell} \otimes \boldsymbol{\Theta}_\ell) \end{bmatrix} \right)$
5. Gaussian likelihood		$\boldsymbol{y} \vert \boldsymbol{\theta}_1, \theta_2 \sim N(\boldsymbol{A}\boldsymbol{\theta}_1, \theta_2 \boldsymbol{I})$

4.1.2. The Inverse Wishart Prior Fragment

We define the *inverse Wishart prior* fragment to correspond to the $d^{\Theta} \times d^{\Theta}$ random matrix $\boldsymbol{\Theta}$ satisfying

$$\boldsymbol{\Theta} \sim \text{Inverse-Wishart}(\kappa_{\boldsymbol{\Theta}}, \boldsymbol{\Lambda}_{\boldsymbol{\Theta}})$$

where $\kappa_{\boldsymbol{\Theta}} > 0$ and $\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}$ is a $d^{\Theta} \times d^{\Theta}$ symmetric positive-definite matrix. This fragment, also shown in Table 1, has factor

$$p(\boldsymbol{\Theta}) = \mathcal{C}^{-1}_{d^{\Theta}, \kappa_{\boldsymbol{\Theta}}} |\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}|^{\kappa_{\boldsymbol{\Theta}}/2} |\boldsymbol{\Theta}|^{-(\kappa_{\boldsymbol{\Theta}} + d^{\Theta} + 1)/2} \times \exp\{-\frac{1}{2}\text{tr}(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}} \boldsymbol{\Theta}^{-1})\},$$

where

$$\mathcal{C}_{d,\kappa} \equiv 2^{d\kappa/2} \pi^{d(d-1)/4} \prod_{j=1}^{d} \Gamma\left(\frac{\kappa + 1 - j}{2}\right), \tag{29}$$

and the single stochastic node $\boldsymbol{\Theta}$, a symmetric and positive-definite $d^{\Theta} \times d^{\Theta}$ matrix. From the natural form of the Inverse Wishart distribution given in Section S.1.7 of the online supplement, the factor to stochastic node message is proportional to $p(\boldsymbol{\Theta})$ and has natural parameter form:

$$m_{p(\boldsymbol{\Theta}) \to \boldsymbol{\Theta}}(\boldsymbol{\Theta}) = \exp\left\{ \begin{bmatrix} \log|\boldsymbol{\Theta}| \\ \text{vec}(\boldsymbol{\Theta}^{-1}) \end{bmatrix}^T \boldsymbol{\eta}_{p(\boldsymbol{\Theta}) \to \boldsymbol{\Theta}} \right\}.$$

The natural parameter vector is a fixed vector that depends only on the hyperparameters:

$$\boldsymbol{\eta}_{p(\boldsymbol{\Theta}) \to \boldsymbol{\Theta}} \longleftarrow \begin{bmatrix} -\frac{1}{2}(\kappa_{\boldsymbol{\Theta}} + d^{\Theta} + 1) \\ -\frac{1}{2}\text{vec}(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}) \end{bmatrix}.$$

4.1.3. Iterated Inverse G-Wishart Fragment

The *iterated Inverse G-Wishart fragment* is shown in Table 1 and corresponds to the conditional distributional specification

$$\boldsymbol{\Theta}_1 \vert \boldsymbol{\Theta}_2 \sim \text{Inverse-G-Wishart}(G, \kappa, \boldsymbol{\Theta}_2^{-1}),$$

where $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ are $d^{\Theta} \times d^{\Theta}$ random matrices, $\kappa > d^{\Theta} - 1$ is deterministic and G is a d^{Θ}-node undirected graph. See Section S.1.7.1 in the online supplement for the definition of the Inverse G-Wishart distribution.

The rationale for this fragment for Bayesian semiparametric regression stems from the family of marginally noninformative covariance matrix priors given in Huang and Wand (2013). In particular, for a $d \times d$ covariance matrix $\boldsymbol{\Sigma}$, their Equation (2) is equivalent to

$$\begin{aligned} \boldsymbol{\Sigma} \vert \boldsymbol{A} &\sim \text{Inverse-Wishart}(\nu + d - 1, \boldsymbol{A}^{-1}), \\ \boldsymbol{A} &\sim \text{Inverse-G-Wishart}\left(G_{\text{diag}}, 1, \frac{1}{\nu} \underset{1 \le k \le K}{\text{diag}}\,(1/A_k^2)\right) \end{aligned} \tag{30}$$

where $\nu, A_1, \ldots, A_K > 0$ are hyperparameters and G_{diag} is defined in Section S.1.7.1 of the online supplement. Setting $d = \nu = 1$ leads to the variance parameter result (13). For $d > 1$ setting $\nu = 2$ has the attraction of imposing Uniform$(-1, 1)$ priors on the correlation parameters in $\boldsymbol{\Sigma}$ (Huang and Wand 2013).

The fragment factor is of the form

$$p(\boldsymbol{\Theta}_1 \vert\, \boldsymbol{\Theta}_2) \propto |\boldsymbol{\Theta}_2|^{-\kappa/2} |\boldsymbol{\Theta}_1|^{-(\kappa + d^{\Theta} + 1)/2} \exp\left\{-\frac{1}{2}\text{tr}(\boldsymbol{\Theta}_1^{-1}\boldsymbol{\Theta}_2^{-1})\right\}.$$

From (8) and (9), the message that $p(\boldsymbol{\Theta}_1 \vert \boldsymbol{\Theta}_2)$ passes to $\boldsymbol{\Theta}_1$ is

$$m_{p(\boldsymbol{\Theta}_1 \vert \boldsymbol{\Theta}_2) \to \boldsymbol{\Theta}_1}(\boldsymbol{\Theta}_1)$$

$$= \exp\left\{\begin{bmatrix}\log|\boldsymbol{\Theta}_1|\\ \text{vec}(\boldsymbol{\Theta}_1^{-1})\end{bmatrix}^T \boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_1}\right\},$$

where

$$\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_1} \longleftarrow \begin{bmatrix}-(\kappa+d^{\Theta}+1)/2\\ -\frac{1}{2}\text{vec}\left(E_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_1}(\boldsymbol{\Theta}_2^{-1})\right)\end{bmatrix} \tag{31}$$

and $E_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_1}$ denotes expectation with respect to the density function formed by normalizing the message product $m_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_2}(\boldsymbol{\Theta}_2)\, m_{\boldsymbol{\Theta}_2\to p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)}(\boldsymbol{\Theta}_2)$. Under the conjugacy assumption that messages passed to $\boldsymbol{\Theta}_2$ from its other neighboring factors are also within the Inverse-G-Wishart family the expectation in (31) is a special case of

$$E(\boldsymbol{X}^{-1}) \quad \text{where} \quad \boldsymbol{X} \sim \text{Inverse-G-Wishart}(G,\kappa,\boldsymbol{\Lambda}) \tag{32}$$

or, equivalently, the mean of a G-Wishart random matrix. Similarly,

$$m_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_2}(\boldsymbol{\Theta}_2) = \exp\left\{\begin{bmatrix}\log|\boldsymbol{\Theta}_2|\\ \text{vec}(\boldsymbol{\Theta}_2^{-1})\end{bmatrix}^T \boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_2}\right\},$$

where

$$\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_2} \longleftarrow \begin{bmatrix}-\kappa/2\\ -\frac{1}{2}\text{vec}\left(E_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_2}(\boldsymbol{\Theta}_1^{-1})\right)\end{bmatrix}$$

and, assuming that all other messages passed to $\boldsymbol{\Theta}_2$ are within the Inverse G-Wishart family, the natural parameter update is also a special case of (32).

For general, undirected graphs (32) can be very complicated (Uhler et al., 2014). However, for important special cases the required expectation admits a simple closed form expression. These cases are discussed next.

The Case of $d^{\Theta}=1$

If $d^{\Theta}=1$, then $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ reduce to variance parameters and results concerning Inverse Chi-Squared random variables apply. The updates become

$$\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_1} \longleftarrow \begin{bmatrix}-\frac{1}{2}(\kappa+2)\\ -\frac{1}{2}\left(\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_2}\right)_1+1\right)\Big/\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_2}\right)_2\end{bmatrix} \tag{33}$$

and

$$\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_2} \longleftarrow \begin{bmatrix}-\frac{1}{2}\kappa\\ -\frac{1}{2}\left(\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1}\right)_1+1\right)\Big/\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1}\right)_2\end{bmatrix}. \tag{34}$$

Note use of the notation first used at (23).

The Case of $d^{\Theta}>1$ and G Totally Connected or Totally Disconnected

If G is a totally connected d-node graph, meaning that there is an edge between each pair of nodes, then the Inverse G-Wishart distribution coincides with the ordinary Inverse Wishart distribution and the well-known result

$$\boldsymbol{X} \sim \text{Inverse-Wishart}(\kappa,\boldsymbol{\Lambda}) \quad \text{implies} \quad E(\boldsymbol{X}^{-1}) = \kappa\boldsymbol{\Lambda}^{-1} \tag{35}$$

applies. Suppose instead that G is totally disconnected, meaning that it has no edges. Then $G = G_{\text{diag}}$ in the notation of Section S.1.7.1 of the online supplement and $\boldsymbol{\Lambda}$ is a diagonal matrix. It is easily established that (35) also applies in the totally disconnected case. Switching to natural parameters via (S.6) of the online supplement we obtain the update expressions

$$\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_1} \longleftarrow \begin{bmatrix}-\frac{1}{2}(\kappa+d^{\Theta}+1)\\ -\frac{1}{2}\left\{\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_2}\right)_1+\frac{d^{\Theta}+1}{2}\right\}\\ \times\text{vec}\left[\left\{\text{vec}^{-1}\left(\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_2}\right)_2\right)\right\}^{-1}\right]\end{bmatrix} \tag{36}$$

and

$$\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\to\boldsymbol{\Theta}_2} \longleftarrow \begin{bmatrix}-\frac{1}{2}\kappa\\ -\frac{1}{2}\left\{\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1}\right)_1+\frac{d^{\Theta}+1}{2}\right\}\\ \times\text{vec}\left[\left\{\text{vec}^{-1}\left(\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1}\right)_2\right)\right\}^{-1}\right]\end{bmatrix}, \tag{37}$$

where

$$\begin{bmatrix}\left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1}\right)_1\\ \left(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1}\right)_2\end{bmatrix}$$

is the partition of $\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1}$ for which $(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1})_1$ is the first entry of the vector and $(\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_1})_2$ contains the remaining entries. Similar partitional notation applies to $\boldsymbol{\eta}_{p(\boldsymbol{\Theta}_1|\boldsymbol{\Theta}_2)\leftrightarrow\boldsymbol{\Theta}_2}$.

The Case of $d^{\Theta}>1$ and G Partially Connected

This case suffers from the fact that (32) does not have a simple expression for general partially connected G. However, the

inverse G-Wishart forms that commonly arise in Bayesian semiparametric regression analysis are covered by the previous cases. Hence, this case can be left aside for common models.

4.1.4. Gaussian Penalization Fragment

The fourth fragment in Table 1 is the *Gaussian penalization fragment* since it imposes Gaussian distributional penalties on random effects parameters. The corresponding conditional distributional specification is

$$\begin{bmatrix}\boldsymbol{\theta}_0\\ \vdots\\ \boldsymbol{\theta}_L\end{bmatrix}\Bigg|\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L \sim N\left(\begin{bmatrix}\boldsymbol{\mu}_{\boldsymbol{\theta}_0}\\ \boldsymbol{0}\end{bmatrix},\right.$$
$$\left.\begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{\theta}_0} & \boldsymbol{O}^T\\ \boldsymbol{O} & \underset{1\le\ell\le L}{\text{blockdiag}}\,(\boldsymbol{I}_{m_\ell}\otimes\boldsymbol{\Theta}_\ell)\end{bmatrix}\right),$$

where $\boldsymbol{O}$ is an appropriately sized matrix of zeroes. The $d_0^\theta\times 1$ vector $\boldsymbol{\theta}_0$ is a fixed effects parameter and has a $d_0^\theta\times 1$ deterministic mean $\boldsymbol{\mu}_{\boldsymbol{\theta}_0}$ and $d_0^\theta\times d_0^\theta$ deterministic covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}_0}$. The covariance matrices $\boldsymbol{\Theta}_\ell$ are stochastic and have dimension $d_\ell^\theta\times d_\ell^\theta$, $1\le\ell\le L$. The random effects vectors $\boldsymbol{\theta}_\ell$ are also stochastic and have dimension $(m_\ell d_\ell^\Theta)\times 1$, $1\le\ell\le L$.

The fragment factor is

$$\begin{aligned}p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L) &= (2\pi)^{-d_0^\theta/2}|\boldsymbol{\Sigma}_{\boldsymbol{\theta}_0}|^{-1/2}\\ &\times\exp\left\{-\frac{1}{2}(\boldsymbol{\theta}_0-\boldsymbol{\mu}_{\boldsymbol{\theta}_0})^T\boldsymbol{\Sigma}_{\boldsymbol{\theta}_0}^{-1}(\boldsymbol{\theta}_0-\boldsymbol{\mu}_{\boldsymbol{\theta}_0})\right\}\\ &\times\prod_{\ell=1}^{L}(2\pi)^{-m_\ell d_\ell^\Theta/2}|\boldsymbol{I}_{m_\ell}\otimes\boldsymbol{\Theta}_\ell|^{-1/2}\\ &\times\exp\Big\{-\frac{1}{2}\boldsymbol{\theta}_\ell^T(\boldsymbol{I}_{m_\ell}\otimes\boldsymbol{\Theta}_\ell^{-1})\boldsymbol{\theta}_\ell\Big\}.\end{aligned}$$

The structure of the fragment is depicted in its diagram in Table 1. We assume that each of $(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L)$ and $\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L$ receive messages from outside the fragment that are conjugate with the message it receives from $p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)$. Update (7) implies that the message from $(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L)$ to the fragment factor is proportional to a Multivariate Normal density function with natural parameter vector $\boldsymbol{\eta}_{(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L)\to p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)}$ and the message from each $\boldsymbol{\Theta}_\ell$ is proportional to an Inverse-G-Wishart density function with natural parameter vector $\boldsymbol{\eta}_{\boldsymbol{\Theta}_\ell\to p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)}$. It follows that the inputs for the Gaussian penalization fragment are

$$\boldsymbol{\eta}_{(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L)\to p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)},$$
$$\boldsymbol{\eta}_{p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)\to(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L)}$$

and

$$\boldsymbol{\eta}_{\boldsymbol{\Theta}_\ell\to p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)}$$
$$\boldsymbol{\eta}_{p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)\to\boldsymbol{\Theta}_\ell},\quad 1\le\ell\le L.$$

Using (8), (9) and Table S.1 in the online supplement, the message from this factor to the coefficient vector $(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L)$ has natural parameter update

$$\boldsymbol{\eta}_{p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)\to(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L)}$$
$$\longleftarrow\begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{\theta}_0}^{-1}\boldsymbol{\mu}_{\boldsymbol{\theta}_0}\\ \boldsymbol{0}\\ -\frac{1}{2}\text{vec}\Big(\text{blockdiag}\left(\boldsymbol{\Sigma}_{\boldsymbol{\theta}_0}^{-1},\underset{1\le\ell\le L}{\text{blockdiag}}\,(\boldsymbol{I}_{m_\ell}\otimes\boldsymbol{\Omega}_\ell)\right)\Big)\end{bmatrix},$$

where $\boldsymbol{0}$ is the $\sum_{\ell=1}^L m_\ell\,d_\ell^\theta\times 1$ vector of zeroes and

$$\begin{aligned}\boldsymbol{\Omega}_\ell\equiv&\left(\left(\boldsymbol{\eta}_{p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)\leftrightarrow\boldsymbol{\Theta}_\ell}\right)_1+\frac{d_\ell^\Theta+1}{2}\right)\\ &\times\left\{\text{vec}^{-1}\left(\left(\boldsymbol{\eta}_{p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)\leftrightarrow\boldsymbol{\Theta}_\ell}\right)_2\right)\right\}^{-1}.\end{aligned}$$

Similarly, the message from $p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)$ to each $\boldsymbol{\Theta}_\ell$, $1\le\ell\le L$, has natural parameter update

$$\boldsymbol{\eta}_{p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)\to\boldsymbol{\Theta}_\ell}\longleftarrow$$
$$\begin{bmatrix}-m_\ell/2\\ G_{\text{VMP}}\left(\boldsymbol{\eta}_{p(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L|\,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_L)\leftrightarrow(\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_L)};\,\boldsymbol{D}_\ell,\boldsymbol{0},0\right)\end{bmatrix}$$

where G_{VMP} is defined by (14) and

$$\begin{aligned}\boldsymbol{D}_\ell\equiv\text{blockdiag}\Bigg(&\boldsymbol{O}_{d_0^\Theta},\boldsymbol{O}_{m_1d_1^\Theta},\ldots,\underset{1\le k\le m_\ell}{\text{blockdiag}}\,(\boldsymbol{J}_{d_\ell^\Theta}),\ldots,\\ &\boldsymbol{O}_{m_Ld_L^\Theta}\Bigg)\end{aligned}$$

with $\boldsymbol{J}_d$ denoting the $d\times d$ matrix with each entry equal to 1 and $\boldsymbol{O}_d$ denoting the $d\times d$ matrix with each entry equal to 0.

While the formulas given in this section cover a wide range of penalization scenarios arising in semiparametric regression we have, with succinctness in mind, left out multilevel models with the number of levels exceeding two. The extension to arbitrarily high levels would take significantly more algebra and obscure the main message regarding the fragment approach. In the same vein, we are not using matrix algebraic streamlining as described in Lee and Wand (2016a, 2016b) for MFVB. Matrix algebraic streamlining is concerned with matters such as avoiding large indicator matrices and redundant calculations. Its extension to VMP would also require significantly more algebra and is left for future research.

4.1.5. Gaussian Likelihood Fragment

The *Gaussian likelihood fragment* corresponds to the form

$$\boldsymbol{y}|\,\boldsymbol{\theta}_1,\theta_2\sim N(\boldsymbol{A}\boldsymbol{\theta}_1,\theta_2\,\boldsymbol{I}),$$

where $\boldsymbol{y}$ is an $n\times 1$ vector of observed data values, and $\boldsymbol{A}$ is an $n\times d^\theta$ design matrix. The stochastic nodes are the $d^\theta\times 1$ coefficient vector $\boldsymbol{\theta}_1$ and the variance parameter $\theta_2>0$. The factor is

$$p(\boldsymbol{y}|\,\boldsymbol{\theta}_1,\theta_2)=(2\pi\theta_2)^{-n/2}\exp\{-(2\theta_2)^{-1}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{\theta}_1\|^2\}.$$

For this fragment, shown in Table 1, we assume that each of the stochastic nodes, $\boldsymbol{\theta}_1$ and θ_2, receive messages from factors outside of the fragment that are conjugate with the message it receives from $p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2)$. Because of (7) this implies that the

message from $\boldsymbol{\theta}_1$ to $p(\boldsymbol{y}|\boldsymbol{\theta}_1, \theta_2)$ is proportional to a Multivariate Normal density function with natural parameter $\boldsymbol{\eta}_{\boldsymbol{\theta}_1 \to p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2)}$ and that from θ_2 to $p(\boldsymbol{y}|\boldsymbol{\theta}_1, \theta_2)$ is proportional to an Inverse Chi-Squared density function with natural parameter $\boldsymbol{\eta}_{\theta_2 \to p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2)}$. It follows that the inputs for the Gaussian likelihood fragment are

$$\boldsymbol{\eta}_{\boldsymbol{\theta}_1 \to p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2)}, \quad \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2) \to \boldsymbol{\theta}_1},$$
$$\boldsymbol{\eta}_{\theta_2 \to p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2)} \quad \text{and} \quad \boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2) \to \theta_2}.$$

The outputs are the following updated natural parameters of the messages passed from $p(\boldsymbol{y}|\boldsymbol{\theta}_1, \theta_2)$ to $\boldsymbol{\theta}_1$ and θ_2:

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2) \to \boldsymbol{\theta}_1} \longleftarrow \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{y} \\ -\frac{1}{2}\text{vec}(\boldsymbol{A}^T\boldsymbol{A}) \end{bmatrix} \times \frac{\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2) \leftrightarrow \theta_2}\right)_1 + 1}{\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2) \leftrightarrow \theta_2}\right)_2} \tag{38}$$

and

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2) \to \theta_2} \longleftarrow \begin{bmatrix} -n/2 \\ G_{\text{VMP}}\left(\boldsymbol{\eta}_{p(\boldsymbol{y}|\boldsymbol{\theta}_1,\theta_2) \leftrightarrow \boldsymbol{\theta}_1}; \boldsymbol{A}^T\boldsymbol{A}, \boldsymbol{A}^T\boldsymbol{y}, \boldsymbol{y}^T\boldsymbol{y}\right) \end{bmatrix} \tag{39}$$

where the notational convention of (22) is followed and G_{VMP} is defined by (14).

4.2. Models Accommodated by the Five Fundamental Fragments

The five fragments covered in Section 4.1 are fundamental to VMP-based Bayesian semiparametric regression and accommodate a wide range of models. Table 2 lists the types of models that can be handled via the Bayesian mixed model-based penalized splines approach to semiparametric regression laid out in Section 2.6.

The models in the left column of Table 2 are part of mainstream semiparametric regression analysis for cross-sectional and grouped data as summarized in, for example, Wood (2006) and Hodges (2013). Chapters 2–9 of Ruppert, Wand, and Carroll (2003) summarize the specific approach taken in the current article. Factor-by-curve interactions are detailed in Coull, Ruppert, and Wand (2001), while Kammann and Wand (2003) described multivariate nonparametric regression and geoadditive models that are in accordance with the VMP fragment set-up of Section 4.1. Similarly, the group-specific curves model of Durban et al. (2005) is accommodated by the Section 4.1 fragments and is illustrated in Section 4.4. Group-specific curve models have a number of alternative formulations (e.g., Donnelly, Laird, and Ware 1995; Verbyla et al. 1999).

Of the five fragments, only the last is specific to Gaussian response semiparametric regression. The other four are applicable to non-Gaussian response models and, when combined with the fragments of Section 5, facilitate handling of a wider range of models such as generalized additive models and generalized linear mixed models.

Table 2. Types of Gaussian response semiparametric regression models that are accommodated by VMP with factor to stochastic node updates as given in Section 4.1 for the five fundamental fragments.

Linear regression	Factor-by-curve interactions
Linear mixed	Varying coefficients
Nonparametric regression	Multivariate nonparametric regression
Additive	Geoadditive
Additive mixed	Group-specific curves

4.3. Coding Issues

According to the VMP approach with fragment identification, the updates of the natural parameters for factor to stochastic node messages only need to be coded once and can be then compartmentalized into functions. Once this is achieved for all fragments present in a particular class of models then the factor to stochastic node messages for a specific model within that class can be handled with calls to these functions. The stochastic node to factor messages are trivial and each requires only a few lines of code.

A more ambitious software project is one that allows the user to specify a model using either syntactic coding rules or a directed acyclic graph drawing interface, such as those used by the BUGS (Lunn et al., 2012), Infer.NET (Minka et al., 2014), VIBES (Bishop, Spiegelhalter, and Winn 2003), and Stan (Stan Development Team 2016) Bayesian inference engines, and then internally construct an appropriate factor graph and perform VMP message updates.

As already discussed, Infer.NET is the main software platform providing support for VMP-based inference for general classes of Bayesian models and its interior architecture makes use of fragment-type rules such as those treated in (4.1) to handle arbitrarily large models that are accommodated by these rules. In Wang and Wand (2011) and Luts et al. (2017), we show that versions of Infer.NET can handle various semiparametric regression models provided that particular "tricks" are used. For example, the conjugacy rules of Infer.NET 2.5, Beta 2 do not allow for the standard auxiliary variable representation of the Laplace distribution (e.g., Equation (4) of Park and Casella 2008) and an alternative approximate representation is used in Section 8 of Luts et al. (2015). More complicated semiparametric regression scenarios such as interactions handled using tensor product splines (e.g., Wood, Scheipl, and Faraway 2013), nonparametric variance function estimation (e.g., Menictas and Wand 2015), streamlined variational inference for longitudinal and multilevel models (e.g., Lee and Wand 2016a, 2016b) and missing data (e.g., Faes, Ormerod, and Wand 2011) require self-implementation and the development of new fragments .

This article is not concerned primarily with coding issues but rather the mathematics of VMP aimed at facilitating personal coding of VMP and development of updating formulas for more elaborate semiparametric regression and other statistical models.

4.4. Illustration for Group-Specific Curves Semiparametric Regression

The five fundamental fragments of Section 4.1 can handle quite complicated models as we now demonstrate for data from a

longitudinal study on adolescent somatic growth, described in detail by Pratt et al. (1989). The main variables are

$$y_{ij} = j\text{th height measurement (centimetres) of subject } i,$$
$$\text{and } x_{ij} = \text{age (years) of subject } i \text{ when } y_{ij} \text{ is recorded,}$$

for $1 \le j \le n_i$ and $1 \le i \le m$. We restrict attention to the males in the study, which results in $m = 116$ subjects. The subjects are categorized into black ethnicity (28 subjects) and white ethnicity (88 subjects) and comparison of mean height between the two populations is of interest. The group-specific curve model takes the form

$$y_{ij} = \begin{cases} f_B(x_{ij}) + g_i(x_{ij}) + \varepsilon_{ij} & \text{for black subjects} \\ f_W(x_{ij}) + g_i(x_{ij}) + \varepsilon_{ij} & \text{for white subjects,} \end{cases}$$

where f_B is the mean height function for the black population, f_W is the mean height function for the white population, the functions g_i, $1 \le i \le m$, represent the deviations from ith subject's mean function, and ε_{ij} is the within-subject random error. The penalized spline models are of the form

$$f_W(x) = \beta_0^{\mathrm{W}} + \beta_1^{\mathrm{W}} x + \sum_{k=1}^{K_{\mathrm{gbl}}} u^W_{\mathrm{gbl},k} z_{\mathrm{gbl},k}(x),$$

$$f_B(x) = \beta_0^{\mathrm{W}} + \beta_0^{\mathrm{BvsW}} + \left(\beta_1^{\mathrm{W}} + \beta_1^{\mathrm{BvsW}}\right) x + \sum_{k=1}^{K_{\mathrm{gbl}}} u^B_{\mathrm{gbl},k} z_{\mathrm{gbl},k}(x)$$

$$\text{and} \quad g_i(x) = U_{0i} + U_{1i} x + \sum_{k=1}^{K_{\mathrm{grp}}} u_{\mathrm{grp},ik} z_{\mathrm{grp},k}(x),$$

where $\{z_{\mathrm{gbl},k} : 1 \le k \le K_{\mathrm{gbl}}\}$ and $\{z_{\mathrm{grp},k} : 1 \le k \le K_{\mathrm{grp}}\}$ are spline bases of size K_{gbl} and K_{grp}. The contrast function is

$$c(x) \equiv f_B(x) - f_W(x) = \beta_0^{\mathrm{BvsW}} + \beta_1^{\mathrm{BvsW}} x + \sum_{k=1}^{K_{\mathrm{gbl}}} (u^B_{\mathrm{gbl},k} - u^W_{\mathrm{gbl},k}) z_{\mathrm{gbl},k}(x). \tag{40}$$

Following the mixed model formulation of Durban et al. (2005) and adopting a Bayesian approach leads to the model

$$\begin{aligned}
\boldsymbol{y} \Big| \boldsymbol{\beta}, \boldsymbol{u}^W_{\mathrm{gbl}}, \boldsymbol{u}^B_{\mathrm{gbl}}, \boldsymbol{U}, \boldsymbol{u}_{\mathrm{grp}}, \sigma_\varepsilon \\
\sim N\left(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}^W_{\mathrm{gbl}}\boldsymbol{u}^W_{\mathrm{gbl}} + \boldsymbol{Z}^B_{\mathrm{gbl}}\boldsymbol{u}^B_{\mathrm{gbl}} + \boldsymbol{Z}_U\boldsymbol{U} \right. \\
\left. + \boldsymbol{Z}_{\mathrm{grp}}\boldsymbol{u}_{\mathrm{grp}}, \sigma_\varepsilon^2 \boldsymbol{I}\right), \\
\boldsymbol{u}^W_{\mathrm{gbl}} \Big| \sigma^W_{\mathrm{gbl}} \sim N\left(\boldsymbol{0}, (\sigma^W_{\mathrm{gbl}})^2\right), \quad \boldsymbol{u}^B_{\mathrm{gbl}} \Big| \sigma^B_{\mathrm{gbl}} \sim N\left(\boldsymbol{0}, \left(\sigma^B_{\mathrm{gbl}}\right)^2\right), \\
\boldsymbol{U} \Big| \boldsymbol{\Sigma} \sim N\left(\boldsymbol{0}, \boldsymbol{I}_m \otimes \boldsymbol{\Sigma}\right), \quad \boldsymbol{u}_{\mathrm{grp}} \Big| \sigma_{\mathrm{grp}} \sim N\left(\boldsymbol{0}, \sigma^2_{\mathrm{grp}}\right), \\
\boldsymbol{\beta} \sim N\left(\boldsymbol{0}, \sigma^2_{\boldsymbol{\beta}} \boldsymbol{I}\right), \\
\left(\sigma^W_{\mathrm{gbl}}\right)^2 \Big| a^W_{\mathrm{gbl}} \sim \text{Inverse-}\chi^2\left(1, 1/a^W_{\mathrm{gbl}}\right), \\
\left(\sigma^B_{\mathrm{gbl}}\right)^2 \Big| a^B_{\mathrm{gbl}} \sim \text{Inverse-}\chi^2\left(1, 1/a^B_{\mathrm{gbl}}\right), \\
\sigma^2_{\mathrm{grp}} | a_{\mathrm{grp}} \sim \text{Inverse-}\chi^2\left(1, 1/a_{\mathrm{grp}}\right), \\
\boldsymbol{\Sigma} | \boldsymbol{A} \sim \text{Inverse-Wishart}(3, \boldsymbol{A}^{-1}), \\
a^W_{\mathrm{gbl}} \sim \text{Inverse-}\chi^2\left(1, 1/A^2_{\mathrm{gbl}}\right), \\
a^B_{\mathrm{gbl}} \sim \text{Inverse-}\chi^2\left(1, 1/A^2_{\mathrm{gbl}}\right), \\
a_{\mathrm{grp}} \sim \text{Inverse-}\chi^2\left(1, 1/A^2_{\mathrm{grp}}\right), \\
\boldsymbol{A} \sim \text{Inverse-G-Wishart}\left(G_{\mathrm{diag}}, 1, \frac{1}{2}\boldsymbol{A}_U^{-2}\right), \\
\sigma^2_\varepsilon | a_\varepsilon \sim \text{Inverse-}\chi^2(1, 1/a_\varepsilon), \quad a_\varepsilon \sim \text{Inverse-}\chi^2(1, 1/A^2_\varepsilon)
\end{aligned} \tag{41}$$

for hyperparameters $\sigma^2_{\boldsymbol{\beta}}$, A_{gbl}, A_{grp} A_ε all positive scalars, $\boldsymbol{A}_U$ a 2×2 positive definite diagonal matrix and G_{diag} is a two-node graph without edges, so that $\boldsymbol{A}$ has off-diagonal entries equaling zero. All distributional notation is given in Section S.1 of the online supplement. The coefficient vectors in (41) are

$$\boldsymbol{\beta} \equiv \begin{bmatrix} \beta_0^W \\ \beta_1^W \\ \beta_0^{\mathrm{BvsW}} \\ \beta_1^{\mathrm{BvsW}} \end{bmatrix}, \quad \boldsymbol{u}^W_{\mathrm{gbl}} \equiv \begin{bmatrix} u^W_{\mathrm{gbl},1} \\ \vdots \\ u^W_{\mathrm{gbl},K_{\mathrm{gbl}}} \end{bmatrix},$$

$$\boldsymbol{U} \equiv \begin{bmatrix} U_{01} \\ U_{11} \\ \vdots \\ U_{0m} \\ U_{1m} \end{bmatrix} \quad \text{and} \quad \boldsymbol{u}_{\mathrm{grp}} \equiv \begin{bmatrix} u_{\mathrm{grp},11} \\ \vdots \\ u_{\mathrm{grp},1K_{\mathrm{grp}}} \\ \vdots \\ u_{\mathrm{grp},m1} \\ \vdots \\ u_{\mathrm{grp},mK_{\mathrm{grp}}} \end{bmatrix}$$

with $\boldsymbol{u}^B_{\mathrm{gbl}}$ defined analogously to $\boldsymbol{u}^W_{\mathrm{gbl}}$. The design matrices $\boldsymbol{X}$, $\boldsymbol{Z}^B_{\mathrm{gbl}}$ and $\boldsymbol{Z}_U$ are

$$\boldsymbol{X} \equiv \begin{bmatrix} \boldsymbol{1} & \boldsymbol{x}_1 & \boldsymbol{I}_1^B & \boldsymbol{I}_1^B \odot \boldsymbol{x}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{1} & \boldsymbol{x}_m & \boldsymbol{I}_m^B & \boldsymbol{I}_m^B \odot \boldsymbol{x}_m \end{bmatrix},$$

$$\boldsymbol{Z}^B_{\mathrm{gbl}} \equiv \begin{bmatrix} \boldsymbol{I}_1^B \odot z_{\mathrm{gbl},1}(\boldsymbol{x}_1) & \cdots & \boldsymbol{I}_1^B \odot z_{\mathrm{gbl},K_{\mathrm{gbl}}}(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \boldsymbol{I}_m^B \odot z_{\mathrm{gbl},1}(\boldsymbol{x}_m) & \cdots & \boldsymbol{I}_m^B \odot z_{\mathrm{gbl},K_{\mathrm{gbl}}}(\boldsymbol{x}_m) \end{bmatrix} \quad \text{and}$$

$$\boldsymbol{Z}_U \equiv \underset{1\le i\le m}{\text{blockdiag}}[\mathbf{1}\ \boldsymbol{x}_i]$$

with $\boldsymbol{x}_i$ equaling the $n_i \times 1$ vector containing the x_{ij}, $1 \le j \le n_i$, and $\boldsymbol{I}^B_i$ equaling the $n_i \times 1$ vector with each entry set to I^B_i with $I^B_i = 1$ if the ith subject is black and $I^B_i = 0$ if the ith subject is white. The matrix $\boldsymbol{Z}^W_{\text{gbl}}$ is defined in a similar manner to $\boldsymbol{Z}^B_{\text{gbl}}$, but with I^B_i replaced by $1 - I^B_i$. The design matrix $\boldsymbol{Z}_{\text{grp}}$ has block diagonal structure similar to $\boldsymbol{Z}_U$ with blocks analogous to $\boldsymbol{Z}^B_{\text{gbl}}$ and $\boldsymbol{Z}^W_{\text{gbl}}$ but there is allowance for a different, typically smaller, spline basis of size K_{grp}. The prior on $\boldsymbol{\Sigma}$, in terms of the auxiliary variable $\boldsymbol{A}$, has entries that are marginally noninformative as explained in Huang and Wand (2013).

Figure 6 shows the factor graph of (41) according to the q-density product restriction

$$\begin{aligned} q\Big(&\boldsymbol{\beta}, \boldsymbol{u}, a_\varepsilon, a^B_{\text{gbl}}, a^W_{\text{gbl}}, a_{\text{grp}}, \boldsymbol{A}, \sigma^2_\varepsilon, \left(\sigma^W_{\text{gbl}}\right)^2, \\ &\left(\sigma^B_{\text{gbl}}\right)^2, \boldsymbol{\Sigma}, \sigma^2_{\text{grp}}\Big) \\ = q(\boldsymbol{\beta}, \boldsymbol{u})\, q\Big(&a_\varepsilon, a^B_{\text{gbl}}, a^W_{\text{gbl}}, a_{\text{grp}}, \boldsymbol{A}, \sigma^2_\varepsilon, \left(\sigma^W_{\text{gbl}}\right)^2, \\ &\left(\sigma^B_{\text{gbl}}\right)^2, \boldsymbol{\Sigma}, \sigma^2_{\text{grp}}\Big) \\ = q(\boldsymbol{\beta}, \boldsymbol{u}) q(a_\varepsilon) q\left(a^B_{\text{gbl}}\right) q\left(a^W_{\text{gbl}}\right) q(a_{\text{grp}}) q(\boldsymbol{A}) q\left(\sigma^2_\varepsilon\right) \\ \times q\left(\left(\sigma^W_{\text{gbl}}\right)^2\right) &q\left(\left(\sigma^B_{\text{gbl}}\right)^2\right) q(\boldsymbol{\Sigma}) q\left(\sigma^2_{\text{grp}}\right). \end{aligned}$$

with the second equality justified by induced factorization theory (e.g., Section 10.2.5 of Bishop 2006).

Notwithstanding the complexity of Figure 6, it is simply a conglomeration of four of the fundamental fragments of Table 1, indicated by the number adjacent to each factor. Therefore the factor to stochastic node messages for VMP-based inference are special cases of the messages given in Section 4.1 and can be updated using the formulas given there. The stochastic node to factor messages have trivial updates based on (7). Running 100 iterations of these updates leads to the fitted group-specific curves for 35 randomly chosen subjects and contrast curve shown in Figure 7. MCMC-based fits, obtained using the R package rstan (Stan Development Team 2016), are also shown for comparison. VMP is seen to be in very good agreement with MCMC. The right panel of Figure 7 shows the estimated height gap between black male adolescents and white male adolescents as a function of age. It is highest and (marginally) statistically significant up to about 14 years of age, peaking at 13 years of age. Between 17 and 20 years old there is no discernible height difference between the two populations.

5. Extension to Generalized Semiparametric Regression

Now, we turn to the situation where the response data are not Gaussian and, in particular, are binary or counts. This corresponds to the *generalized* extension of linear models. In the same vein, *generalized linear mixed models* and *generalized additive models* are extensions of models treated in Section 3 that fall under the umbrella of *generalized semiparametric regression*. Viable VMP algorithms for generalized semiparametric regression need to be developed on a case-to-case basis. In this section, we treat binary response semiparametric regression, with both logistic and probit link functions, and Poisson semiparametric regression.

The logistic case is handled here using the variational lower bound of Jaakkola and Jordan (2000). In the probit case, a rather different approach is used based on the auxiliary variable representation of Albert and Chib (1993). Girolami and Rogers (2006) and Consonni and Marin (2007) show how the Albert–Chib device results in tractable MFVB algorithms for probit models. The Poisson case uses yet another approach based on the nonconjugate VMP infrastructure laid out in Knowles and Minka (2011) and the fully simplified Multivariate Normal updates derived in Wand (2014). Knowles and Minka (2011) and Tan and Nott (2013) also proposed quadrature-based approaches for handling the logistic case, but are not investigated here.

The beauty of the VMP approach is that only messages passed between fragments near the likelihood part of the factor graph are affected by a change from the Gaussian response situation to each of these generalized response situations. Figure 8 shows the fragments involved. The left panel diagram of Figure 8 is appropriate for both logistic models handled via the Jaakkola and Jordan (2000) approach and Poisson response models handled via the Knowles and Minka (2011) approach with the Wand (2014) updates. The fragment is called the *Jaakkola–Jordan*

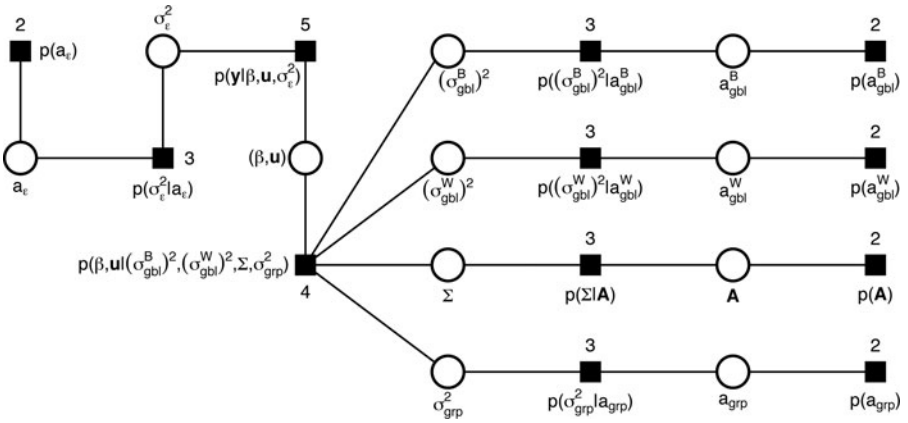


Figure 6. Factor graph corresponding to the group specific curve model (41). The number adjacent to each factor signifies the fragment number in Table 1.

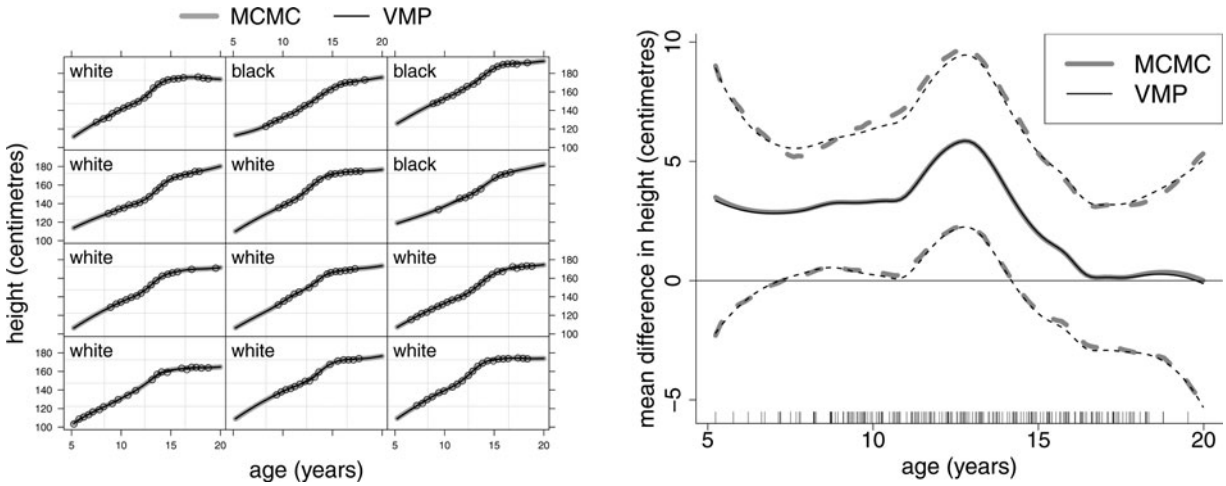


Figure 7. Left panel: comparison of MCMC-based and VMP-based fitted group-specific curves for 12 randomly chosen subjects from the data on adolescent somatic growth (Pratt et al. 1989). The legend in each panel signifies the subject's ethnicity. Right panel: similar to the left panel but for the estimated contrast curve. The dashed curves correspond to approximate pointwise 95% credible intervals. The tick marks at the base of the plot show the age data.

logistic fragment or the *Knowles-Minka-Wand fragment* depending on the response type. In Sections 5.1 and 5.3, we provide analytic updating formulas for the sufficient statistic of $m_{p(\boldsymbol{y}|\boldsymbol{\theta})\to\boldsymbol{\theta}}(\boldsymbol{\theta})$ assuming that Multivariate Normal messages are being passed to and from the $\boldsymbol{\theta}$ stochastic node.

Throughout this section $\boldsymbol{\theta}$ denotes an $d \times 1$ random vector and $\boldsymbol{A}$ denotes an $n \times d$ design matrix.

5.1. Jaakkola–Jordan Updates for the Logistic Likelihood Fragment

The *logistic fragment* is concerned with the logistic likelihood specification

$$y_i|\boldsymbol{\theta} \overset{\text{ind.}}{\sim} \text{Bernoulli}\big(1/[1+\exp\{-(\boldsymbol{A}\boldsymbol{\theta})_i\}]\big), \quad 1 \le i \le n.$$

The factor of this fragment is

$$p(\boldsymbol{y}|\,\boldsymbol{\theta}) = \exp\Big[\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{1}^T\log\{\boldsymbol{1}+\exp(\boldsymbol{A}\boldsymbol{\theta})\}\Big]. \tag{42}$$

Based on inputs

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\to\boldsymbol{\theta}} \quad \text{and} \quad \boldsymbol{\eta}_{\boldsymbol{\theta}\to p(\boldsymbol{y}|\,\boldsymbol{\theta})}$$

the variational and natural parameter vectors have the following updates:

$$\boldsymbol{\Xi} \longleftarrow \frac{1}{4}\Big\{\text{vec}^{-1}\Big(\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_2\Big)\Big\}^{-1}$$

$$\times\Big[\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_1\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_1^T$$

$$\times\Big\{\text{vec}^{-1}\Big(\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_2\Big)\Big\}^{-1} - 2\boldsymbol{I}\Big],$$

$$\boldsymbol{\xi} \longleftarrow \sqrt{\text{diagonal}\,(\boldsymbol{A}\,\boldsymbol{\Xi}\,\boldsymbol{A}^T)},$$

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\to\boldsymbol{\theta}} \longleftarrow \begin{bmatrix} \boldsymbol{A}^T(\boldsymbol{y}-\frac{1}{2}\boldsymbol{1}) \\ -\text{vec}\left(\boldsymbol{A}^T\text{diag}\left\{\dfrac{\tanh(\boldsymbol{\xi}/2)}{4\boldsymbol{\xi}}\right\}\boldsymbol{A}\right)\end{bmatrix}. \tag{43}$$

Justification for these updates is given in Section S.2.2 of the online supplement.

5.2. Updates for the Albert–Chib Probit Likelihood Fragments

The *Albert–Chib probit fragments* deal with the probit likelihood specification

$$y_i|\boldsymbol{\theta} \overset{\text{ind.}}{\sim} \text{Bernoulli}\Big(\Phi\{(\boldsymbol{A}\boldsymbol{\theta})_i\}\Big), \quad 1 \le i \le n. \tag{44}$$

where Φ is the $N(0,1)$ cumulative distribution function. Following Albert and Chib (1993), we rewrite (44) as

$$y_i|a_i \overset{\text{ind.}}{\sim} \text{Bernoulli}\Big(I(a_i \ge 0)\Big), \qquad 1 \le i \le n,$$

$$\boldsymbol{a}|\boldsymbol{\theta} \sim N(\boldsymbol{A}\boldsymbol{\theta}, \boldsymbol{I}) \tag{45}$$

and work with the factor graph fragments shown in the right panel of Figure 8.

Based on the inputs $\boldsymbol{\eta}_{p(\boldsymbol{a}|\,\boldsymbol{\theta})\to\boldsymbol{\theta}}$ and $\boldsymbol{\eta}_{\boldsymbol{\theta}\to p(\boldsymbol{a}|\,\boldsymbol{\theta})}$, the updates for the Albert-Chib probit fragments are

$$\boldsymbol{v} \longleftarrow -\frac{1}{2}\boldsymbol{A}\Big\{\text{vec}^{-1}\Big(\big(\boldsymbol{\eta}_{p(\boldsymbol{a}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_2\Big)\Big\}^{-1}$$

$$\times\big(\boldsymbol{\eta}_{p(\boldsymbol{a}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_1,$$

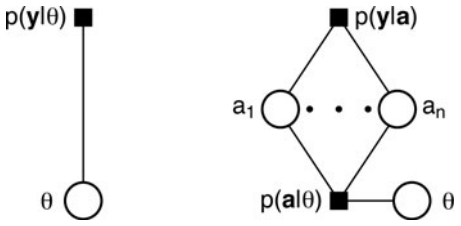


Figure 8. Left panel: diagram for fragment corresponding to the likelihood for logistic and Poisson regression models. Right panel: fragments for the likelihood for probit regression models with auxiliary variables $a_1, \ldots, a_n$ corresponding to the Albert–Chib device.

$$\boldsymbol{\eta}_{p(\boldsymbol{a}|\,\boldsymbol{\theta})\to\boldsymbol{\theta}} \longleftarrow \begin{bmatrix} \boldsymbol{A}^T\left\{\boldsymbol{v} + (2\boldsymbol{y}-1)\odot\zeta'\Big((2\boldsymbol{y}-1)\odot\boldsymbol{v}\Big)\right\} \\ -\frac{1}{2}\text{vec}(\boldsymbol{A}^T\boldsymbol{A}) \end{bmatrix} \tag{46}$$

where

$$\zeta(x) \equiv \log\{2\Phi(x)\} \quad \text{implying that} \quad \zeta'(x) = \frac{(2\pi)^{-1/2}e^{-x^2/2}}{\Phi(x)}.$$

Working with ζ' has the advantage that software, such as the function `zeta()` in the package sn (Azzalini 2015) within the R computing environment (R Core Team 2015), that facilitates numerically stable computation of (46).

Justification for these updates is given in Section S.2.3 of the online supplement.

5.3. Knowles–Minka–Wand Updates for the Poisson Likelihood Fragment

The generic Poisson regression likelihood is

$$y_i|\boldsymbol{\theta} \sim \text{Poisson}\left(\exp\{(\boldsymbol{A}\boldsymbol{\theta})_i\}\right), \quad 1 \le i \le n.$$

Based on inputs

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\to\boldsymbol{\theta}} \quad \text{and} \quad \boldsymbol{\eta}_{\boldsymbol{\theta}\to p(\boldsymbol{y}|\,\boldsymbol{\theta})},$$

the update of $\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\to\boldsymbol{\theta}}$ involves the steps

$$\boldsymbol{\omega} \longleftarrow \exp\left(-\frac{1}{2}\boldsymbol{A}\left\{\text{vec}^{-1}\Big(\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_2\Big)\right\}^{-1}\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_1 - \frac{1}{4}\text{diagonal}\left[\boldsymbol{A}\left\{\text{vec}^{-1}\Big(\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_2\Big)\right\}^{-1}\boldsymbol{A}^T\right]\right)$$

$$\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\to\boldsymbol{\theta}} \longleftarrow \begin{bmatrix} \boldsymbol{A}^T\Big[\boldsymbol{y}-\boldsymbol{\omega} \\ -\frac{1}{2}\text{diag}\,(\boldsymbol{\omega})\boldsymbol{A}\left\{\text{vec}^{-1}\Big(\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_2\Big)\right\}^{-1}\big(\boldsymbol{\eta}_{p(\boldsymbol{y}|\,\boldsymbol{\theta})\leftrightarrow\boldsymbol{\theta}}\big)_1\Big] \\ -\frac{1}{2}\text{vec}\big(\boldsymbol{A}^T\text{diag}\,(\boldsymbol{\omega})\boldsymbol{A}\big) \end{bmatrix}. \tag{47}$$

Full justification of (47) is given in Section S.2.4 of the online supplement. Note that, despite their involved form, the manipulations required to update the factor to stochastic node message are purely algebraic. Again we point out that, according to the message passing approach, (47) only needs to be implemented once when developing a suite of computer programs for VMP-based semiparametric regression.

5.4. Illustration for Generalized Response Nonparametric Regression

We now provide brief illustration of the fragments presented in this section for nonparametric regression via mixed model-based penalized splines with synthetic data. Accuracy compared with MCMC-based inference is also addressed. A timing comparison is given in Section 6.

A sample of size 500 was generated from the Uniform distribution on (0, 1), which we denote by $x_1, \ldots, x_{500}$ and then binary and count responses were generated according to

$$y_i^b|x_i \overset{\text{ind.}}{\sim} \text{Bernoulli}\{f_{\text{true}}(x_i)\} \quad \text{and}$$

$$y_i^c|x_i \overset{\text{ind.}}{\sim} \text{Poisson}\{10\,f_{\text{true}}(x_i)\}, \quad 1 \le i \le 500,$$

where $f_{\text{true}}(x) \equiv \{1.05 - 1.02\,x + 0.018\,x^2 + 0.4\,\phi(x; 0.38, 0.08) + 0.08\,\phi(x; 0.75, 0.03)\}/2.7$ and $\phi(x; \mu, \sigma) \equiv (2\pi\sigma^2)^{-1/2}\exp\{-\frac{1}{2}(x-\mu)/\sigma^2\}$. The logistic, probit, and Poisson penalized spline models for the mean functions take the forms

$$H\left(\beta_0 + \beta_1\,x + \sum_{k=1}^{K} u_k\,z_k(x)\right), \quad u_k \overset{\text{ind.}}{\sim} N(0, \sigma_u^2)$$

where, respectively, $H(x) = 1/(1+e^{-x})$, $H(x) = \Phi^{-1}(x)$ and $H(x) = e^x$ and the z_k are spline basis functions as defined just after (12). The priors $\beta_0, \beta_1 \overset{\text{ind.}}{\sim} N(0, 10^{10})$ and $\sigma_u \sim$ Half-Cauchy(10^5) were imposed. A canonical cubic O'sullivan spline basis with $K = 25$ was used for the z_k, formed by placing the interior knots at quantiles of the x_is. MCMC samples from the posterior distributions of the coefficients of size 1000, after a warm-up of 1000, were obtained using the R package rstan (Stan Development Team 2016). VMP fitting is similar to the updating scheme described in Section 3.2 but with likelihood fragment updating steps described in Sections 5.1–5.3 rather than those for the Gaussian likelihood fragment and was iterated 200 times for each model.

Figure 9 displays the true mean function and the MCMC and VMP fits. In the logistic and Poisson models it is difficult to discern a difference between the posterior means and pointwise 95% credible intervals. The probit fits are such that VMP gives credible sets that are slightly too narrow. This shortcoming of MFVB/VMP for the Albert–Chib probit approach is attributable to posterior correlations between the entries $\boldsymbol{a}$ and those of $(\beta_0, \boldsymbol{u})$ conveniently being set to zero in the mean field approximation even though these correlations are significantly nonzero (e.g., Holmes and Held 2006).

6. Speed Considerations

As the title of this article indicates, the MFVB/VMP approach offers fast approximate inference. For models of reasonable size, fits can be achieved in a few seconds or less on ordinary desktop and laptop computers. Seven of the author's previously published MFVB articles contain speed comparisons with MCMC including Faes, Ormerod, and Wand (2011), Lee and Wand (2016b), and Luts and Wand (2015) out of those which are referenced earlier. The speed advantages for the VMP alternative also apply, although some qualification is necessary due to whether or not matrix algebraic streamlining is employed. In Table 1 of Lee and Wand (2016a), it is shown that MFVB/VMP fitting of large semiparametric longitudinal and multilevel models with matrix algebraic streamlining and low-level programming language implementation can be achieved in seconds even when there are tens of thousands of groups. A similar story is told by Table 1 of Lee and Wand (2016b) for large to very large group-specific curve models. The MFVB/VMP computing times range

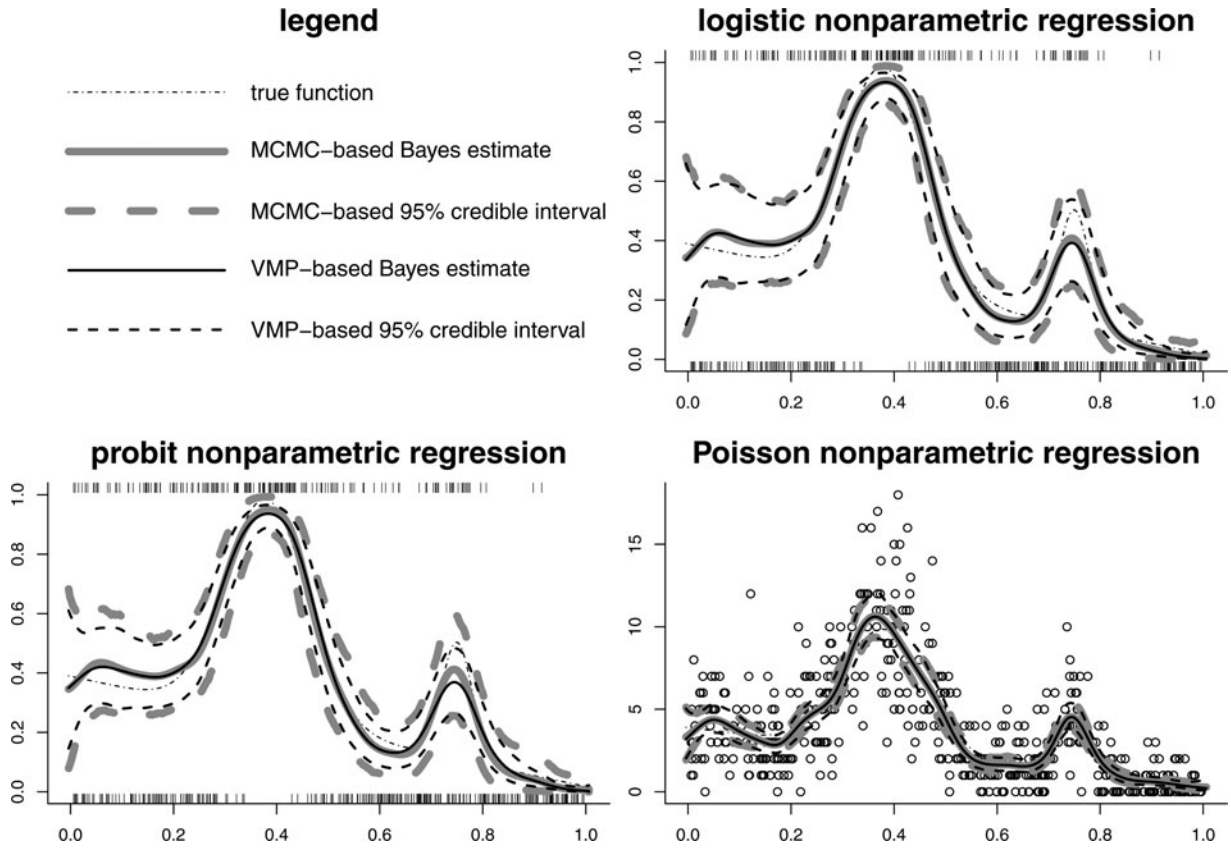


Figure 9. Comparison of MCMC- and VMP-based inference for simulated Bernoulli and Poisson response nonparametric regression data. The solid curves are Bayes estimates whilst the dash curves correspond to pointwise 95% credible intervals. The tick marks at the top and base of the binary response plots show the data.

from minutes to tens of minutes for the largest models considered, although this is without low-level programming language implementation. It is stated that MCMC fitting for the same models is expected to take days to weeks to run.

Table 3 shows the average and standard deviation of computing times in seconds for replications of Figure 9 simulation example. All computations were performed on a laptop computer with 8 GB of random access memory and a 1.7 GHz processor. There are a number of caveats connected with Table 3: (a) the computing times depend on the MCMC sample sizes and the number of VMP iterations, (b) the MCMC iterations are performed by Stan in the faster low-level C++ programming language whereas the VMP iterations are performed in the slower high-level R programming language, (c) the VMP approach was implemented naïvely using the formulas of Section 4.1.4 without any matrix algebraic streamlining. Each of these caveats disadvantage the VMP approach in the speed comparison. Nevertheless, Table 3 shows that VMP takes 1.5 seconds or less to perform approximate Bayesian inference for Figure 9 scatterplots, whereas close to a minute is needed for MCMC via Stan.

Table 3. Average (standard deviation) of computational times in seconds over 100 replications of Figure 9 simulated data example.

Method	Logistic nonpar. reg'n	Probit nonpar. reg'n	Poisson nonpar. reg'n
MCMC	49.50 (6.85)	56.200 (7.56)	48.60 (4.340)
VMP	1.36 (0.117)	0.327 (0.0307)	1.52 (0.101)

7. Conclusion

We have demonstrated that approximate inference for particular classes of arbitrarily large semiparametric regression models can be implemented with relatively few computer code compartments. Moreover, many of these compartments involve straightforward matrix algebraic manipulations. Our exposition transcends ongoing software projects that make use of MFVB/VMP. Extensions to more elaborate models within the VMP framework is elucidated.

Accuracy considerations aside, the algebraic infrastructure that we have laid out in this article has far-reaching implications for the analysis of big datasets via large semiparametric models as both continue to grow in size. It is also beneficial for other classes of statistical models. In situations where inferential accuracy is paramount, variational message passing algorithms may still play important roles in design and model selection phases with final reporting based on a more accurate method.

Supplementary Materials

The online supplementary materials provide technicalities including detailed derivations and justifications.

Acknowledgments

This research was partially supported by the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers. The author thanks Ray Carroll, Peter Forrester, Andy Kim, Cathy Lee, Matt McLean, Marianne Menictas, Tui Nolan, Chris Oates, and Donald Richards for their

comments on this research. Comments from an associate editor and three referees are also gratefully acknowledged.

ORCID

M. P. Wand http://orcid.org/0000-0003-2555-896X

References

Albert, J.H., and Chib, S. (1993), "Bayesian Analysis of Binary and Polychotomous Response Data," *Journal of the American Statistical Association*, 88, 669–679. [151,152]

Azzalini, A. (2015), The R package: The skew-normal and skew-*t* distributions (version 1.2), available at *http://azzalini.stat.unipd.it/SN*. [153]

Bishop, C. M., Spiegelhalter, D. J., and Winn, J. (2003), "VIBES: A Variational Inference Engine for Bayesian Networks," in *Advances in Neural Information Processing Systems*, eds. S. Becker, S. Thrun and K. Obermayer, pp. 793–800, Cambridge, MA: MIT Press. [149]

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, New York: Springer. [141,144,151]

Consonni, G., and Marin, J.-M. (2007), "Mean-Field Variational Approximate Bayesian Inference for Latent Variable Models," *Computational Statistics and Data Analysis*, 52, 790–798. [151]

Coull, B. A., Ruppert, D., and Wand, M. P. (2001), "Simple Incorporation of Interactions into Additive Models," *Biometrics*, 57, 539–545. [149]

Diggle, P., Heagerty, P., Liang, K.-L., and Zeger, S. (2002), *Analysis of Longitudinal Data* (2nd ed.), Oxford, UK: Oxford University Press. [141]

Donnelly, C. A., Laird, N. M. and Ware, J. H. (1995), "Prediction and Creation of Smooth Curves for Temporally Correlated Longitudinal Data," *Journal of the American Statistical Association*, 90, 984–989. [149]

Durban, M., Harezlak, J., Wand, M. P., and Carroll, R. J. (2005), "Simple Fitting of Subject-Specific Curves for Longitudinal Data," *Statistics in Medicine*, 24, 1153–1167. [149,150]

Faes, C., Ormerod, J. T., and Wand, M. P. (2011), "Variational Bayesian Inference for Parametric and Nonparametric Regression with Missing Data," *Journal of the American Statistical Association*, 106, 959–971. [138,149,153]

Fitzmaurice, G., Davidian, M., Verbeke, G., and Molenberghs, G. (eds.) (2008), *Longitudinal Data Analysis: A Handbook of Modern Statistical Methods*, Boca Raton, FL: CRC Press. [141]

Frey, B. J., Kschischang, F. R., Loeliger, H. A., and Wiberg, N. (1998), "Factor Graphs and Algorithms," in *Proceedings of the 35th Allerton Conference on Communication, Control and Computing 1997*. [137]

Gelman, A. (2006), "Prior Distributions for Variance Parameters in Hierarchical Models," *Bayesian Analysis*, 1, 515–533. [140]

Gelman, A., and Hill, J. (2007), *Data Analysis using Regression and Multilevel/Hierarchical Models*, New York: Cambridge University Press. [141]

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014), *Bayesian Data Analysis* (3rd ed.), Boca Raton, FL: CRC Press. [141]

Ghosh, S. (2015), *Distributed Systems: An Algorithmic Approach* (2nd ed.), Boca Raton, Florida: CRC Press. [137]

Girolami, M., and Rogers, S. (2006), "Variational Bayesian Multinomial Probit Regression," *Neural Computation*, 18, 1790–1817. [151]

Goldstein, H. (2010), *Multilevel Statistical Models* (4th ed.), Chichester, UK: Wiley. [141]

Gurrin, L. C., Scurrah, K. J., and Hazelton, M. L. (2005), "Tutorial in Biostatistics: Spline Smoothing With Linear Mixed Models," *Statistics in Medicine*, 24, 3361–3381. [140]

Hodges, J. S. (2013), *Richly Parameterized Linear Models: Additive, Time Series, and Spatial Models Using Random Effects*, Boca Raton, FL: CRC Press. [149]

Holmes, C. C., and Held, L. (2006), "Bayesian Auxiliary Variable Models for Binary and Multinomial Regression," *Bayesian Analysis*, 1, 145–168. [153]

Huang, A., and Wand, M. P. (2013), "Simple Marginally Noninformative Prior Distributions for Covariance Matrices," *Bayesian Analysis*, 8, 439–452. [140,146,151]

Jaakkola, T. S., and Jordan, M. I. (2000), "Bayesian Parameter Estimation Via Variational Methods," *Statistics and Computing*, 10, 25–37. [151]

Jordan, M. I. (2004), "Graphical Models," *Statistical Science*, 19, 140–155. [137]

Kammann, E. E., and Wand, M. P. (2003), "Geoadditive Models," *Journal of the Royal Statistical Society,* Series C, 52, 1–18. [149]

Knowles, D. A., and Minka, T. P. (2011), "Non-Conjugate Message Passing for Multinomial and Binary Regression," in *Advances in Neural Information Processing Systems*, eds. J. Shawe-Taylor, R. S. Zamel, P. Bartlett, F. Pereira, and K. Q. Weinberger (Vol. 24), pp. 1701–1709. Cambridge, MA: MIT Press. [151]

Kschischang, F. R., Frey, B. J., and Loeliger, H. A. (2001), "Factor Graphs and the Sum–Product Algorithm," *IEEE Transactions of Information Theory*, 47, 498–519. [137]

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2016), "Automatic Variational Inference in Stan," unpublished manuscript (*arXiv:1603.00788v1*). [138]

Lee, C. Y. Y., and Wand, M. P. (2016a), "Streamlined Mean Field Variational Bayes for Longitudinal and Multilevel Data Analysis," *Biometrical Journal*, 58, 868–895. [148,149,153]

——— (2016b), "Variational Inference for Fitting Complex Bayesian Mixed Effects Models to Health Data," *Statistics in Medicine*, 35, 165–188. [148,149,153]

Lock, R. H. (1993), "1993 New Car Data," *Journal of Statistics Education*, 1. [143,144]

Lunn, D., Jackson, C., Best, N., Thomas, A., and Spiegelhalter, D. (2012), *The BUGS Book – A Practical Introduction to Bayesian Analysis*, Boca Raton, FL: CRC Press. [149]

Luts, J. (2015), "Real-time Semiparametric Regression for Distributed Datasets," *IEEE Transactions on Knowledge and Data Engineering*, 27, 545–557. [138]

Luts, J., Broderick, T., and Wand, M. P. (2014), "Real-Time Semiparametric Regression," *Journal of Computational and Graphical Statistics*, 23, 589–615. [137,138]

Luts, J., and Wand, M. P. (2015), "Variational Inference for Count Response Semiparametric Regression," *Bayesian Analysis*, 10, 991–1023. [138,153]

Luts, J., Wang, S. S. J., Ormerod, J. T., and Wand, M. P. (2017), "Semiparametric Regression Analysis Via Infer.NET," *Journal of Statistical Software*, in press. [149]

Marley, J. K., and Wand, M. P. (2010), "Non-Standard Semiparametric Regression Via BRugs," *Journal of Statistical Software*, 37, 5, 1–30. [137]

Menictas, M., and Wand, M. P. (2015), "Variational Inference for Heteroscedastic Semiparametric Regression," *Australian and New Zealand Journal of Statistics*, 57, 119–138. [149]

Minka, T. (2005), "Divergence Measures and Message Passing," *Microsoft Research Technical Report Series*, MSR-TR-2005-173, 1–17. [137,138,140]

Minka, T., and Winn, J. (2008), "Gates: A Graphical Notation for Mixture Models," *Microsoft Research Technical Report Series*, MSR-TR-2008-185, 1–16. [140]

Minka, T., Winn, J., Guiver, J., Webster, S., Zaykov, Y., Yangel, B., Spengler, A., and Bronskill, J. (2014), Infer.NET 2.6, Microsoft Research Cambridge. Available at *http://research.microsoft.com/infernet*. [138,149]

Ormerod, J. T., and Wand, M. P. (2010), "Explaining Variational Approximations," *The American Statistician*, 64, 140–153. [141]

Park, T., and Casella, G. (2008), "The Bayesian Lasso," *Journal of the American Statistical Association*, 103, 681–686. [149]

Pratt, J. H., Jones, J. J., Miller, J. Z., Wagner, M. A., and Fineberg, N. S. (1989), "Racial Differences in Aldosterone Excretion and Plasma Aldosterone Concentrations in Children," *New England Journal of Medicine*, 321, 1152–1157. [150]

R Core Team (2015), R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Available at *http://www.R-project.org*. [153]

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, New York: Cambridge University Press. [140,149]

Ruppert, D., Wand, M. P., and Carroll, R. J. (2009), "Semiparametric Regression During 2003-2007," *Electronic Journal of Statistics*, 3, 1193–1256. [137]

Stan Development Team (2016), "Stan: A C++ Library for Probability and Sampling, (version 2.9.0)," available at *http://mc-stan.org*. [138,149,151,153]
Tan, L. S. L., and Nott, D. J. (2013), "Variational Inference for Generalized Linear Mixed Models Using Partially Noncentered Parametrizations," *Statistical Science*, 28, 168–188. [151]
Uhler, C., Lenkoski, A., and Richards, D. (2014), "Exact Formulas for the Normalizing Constants of Wishart Distributions for Graphical Models," unpublished manuscript (*arXiv:1406.490*). [147]
Verbyla, A. P., Cullis, B. R., Kenward, M. G., and Welham, S. J. (1999), "The Analysis of Designed Experiments and Longitudinal Data by Using Smoothing Splines" (with discussion), *Applied Statistics*, 48, 269–312. [149]
Wainwright, M. J., and Jordan, M. I. (2008), "Graphical Models, Exponential Families and Variational Inference," *Foundations and Trends in Machine Learning*, 1, 1–305. [141]
Wand, M. P. (2009), "Semiparametric Regression and Graphical Models," *Australian and New Zealand Journal of Statistics*, 51, 9–41. [137,140]
Wand, M. P. (2014), "Fully Simplified multivariate normal updates in Non-Conjugate Variational Message Passing," *Journal of Machine Learning Research*, 15, 1351–1369. [151]
Wand, M. P., and Ormerod, J. T. (2008), "On Semiparametric Regression with O'sullivan Penalized Splines," *Australian and New Zealand Journal of Statistics*, 50, 179–198. [140]
Wand, M. P., and Ormerod, J. T. (2011), "Penalized Wavelets: Embedding Wavelets into Semiparametric Regression," *Electronic Journal of Statistics*, 5, 1654–1717. [137,140]
Wang, S. S. J., and Wand, M. P. (2011), "Using Infer.NET for Statistical Analyses," *The American Statistician*, 65, 115–126. [149]
Winn, J., and Bishop, C. M. (2005), "Variational Message Passing," *Journal of Machine Learning Research*, 6, 661–694. [137,138,140]
Wood, S. N. (2006), *Generalized Additive Models: An Introduction with R*, Boca Raton, FL: CRC Press. [149]
Wood, S. N., Scheipl, F., and Faraway, F. F. (2013), "Straightforward Intermediate Rank Tensor Product Smoothing in Mixed Models," *Statistics and Computing*, 23, 341–3601. [149]

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2017, VOL. 112, NO. 517, Theory and Methods
http://dx.doi.org/10.1080/01621459.2016.1270044

Comment

Dustin Tran and David M. Blei

Department of Computer Science and Statistics, Columbia University, New York, NY

We commend Wand (2017) for an excellent description of message passing (MP) and for developing it to infer large semiparametric regression models. We agree with the author in fully embracing the modular nature of message passing, where one can define "fragments" that enables us to compose localized algorithms. We believe this perspective can aid in the development of new algorithms for automated inference.

Automated inference. The promise of automated algorithms is that modeling and inference can be separated. A user can construct large, complicated models in accordance with the assumptions he or she is willing to make about their data. Then the user can use generic inference algorithms as a computational backend in a "probabilistic programming language," that is, a language for specifying generative probability models.

With probabilistic programming, the user no longer has to write their own algorithms, which may require tedious model-specific derivations and implementations. In the same spirit, the user no longer has to bottleneck their modeling choices to fit the requirements of an existing model-specific algorithm. Automated inference enables probabilistic programming systems, such as Stan (Carpenter et al. 2016), through methods like automatic differentiation variational inference (ADVI; Kucukelbir et al. 2016) and no U-turn sampler (NUTS; Hoffman and Gelman 2014).

Though they aim to apply to a large class of models, automated inference algorithms typically need to incorporate modeling structure to remain practical. For example, Stan assumes that one can at least take gradients of a model's joint density. (Contrast this with other languages that assume one can only sample from the model.) However, more structure is often necessary: ADVI and NUTS are not fast enough by themselves to infer very large models, such as hierarchical models with many groups.

We believe MP and Wand's work could offer fruitful avenues for expanding the frontiers of automated inference. From our perspective, a core principle underlying MP is to leverage structure when it is available—in particular, statistical properties in the model—which provides useful computational properties. In MP, two examples are conditional independence and conditional conjugacy.

From conditional independence to distributed computation. As Wand (2017) indicated, a crucial advantage of message passing is that it modularizes inference; the computation can be performed separately over conditionally independent posterior factors. By definition, conditional independence separates a posterior factor from the rest of the model, which enables MP to define a series of iterative updates. These updates can be run asynchronously and in a distributed environment.

We are motivated by hierarchical models, which substantially benefit from this property. Formally, let y_{nk} be the nth data point in group k, with a total of N_k data points in group k and K many groups. We model the data using local latent variables α_k associated with a group k, and using global latent variables ϕ, which are shared across groups. The model is depicted in Figure 1.

CONTACT David Blei *david.blei@columbia.edu* Department of Computer Science and Statistics, Columbia University, New York, NY 10027, USA.

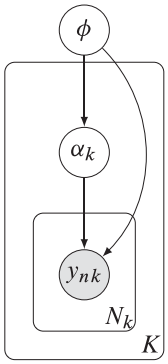


Figure 1. A hierarchical model, with latent variables α_k defined locally per group and latent variables ϕ defined globally to be shared across groups.

The posterior distribution of local variables α_k and global variables ϕ is

$$p(\alpha, \phi|y) \propto p(\phi|y) \prod_{k=1}^{K} \left[p(\alpha_k \mid \phi) \prod_{n=1}^{N_K} p(y_{nk} \mid \alpha_k, \phi) \right].$$

The benefit of distributed updates over the independent factors is immediate. For example, suppose the data consist of 1000 data points per group (with 5000 groups); we model it with 2 latent variables per group and 20 global latent variables. Passing messages, or inferential updates, in parallel provides an attractive approach to handling all 10,020 latent dimensions. (In contrast, consider a sequential algorithm that requires taking 10,019 steps for all other variables before repeating an update of the first.)

While this approach to leveraging conditional independence is straightforward from the message passing perspective, it is not necessarily immediate from other perspectives. For example, the statistics literature has only recently come to similar ideas, motivated by scaling up Markov chain Monte Carlo using divide and conquer strategies (Huang and Gelman 2005; Wang and Dunson 2013). These first analyze data locally over a partition of the joint density, and second aggregate the local inferences. In our work in Gelman et al. (2014), we arrive at the continuation of this idea. Like message passing, the process is iterated, so that local information propagates to global information and global information propagates to local information. In doing so, we obtain a scalable approach to Monte Carlo inference, both from a top-down view, which deals with fitting statistical models to large datasets and from a bottom-up view, which deals with combining information across local sources of data and models.

From conditional conjugacy to exact iterative updates. Another important element of message passing algorithms is conditional conjugacy, which lets us easily calculate the exact distribution for a posterior factor conditional on other latent variables. This enables analytically tractable messages (see Eqs. 7 and 8 of Wand 2017).

Consider the same hierarchical model discussed above, and set

$$p(y_k, \alpha_k \mid \phi) = h(y_k, \alpha_k) \exp\left\{\phi^\top t(y_k, \alpha_k) - a(\phi)\right\},$$
$$p(\phi) = h(\phi) \exp\left\{\eta^{(0)\top} t(\phi) - a(\eta_0)\right\}.$$

The local factor $p(y_k, \alpha_k \mid \phi)$ has sufficient statistics $t(y_k, \alpha_k)$ and natural parameters given by the global latent variable ϕ. The global factor $p(\phi)$ has sufficient statistics $t(\phi) = (\phi, -a(\phi))$, and with fixed hyperparameters $\eta^{(0)}$, which has two components: $\eta^{(0)} = (\eta_1^{(0)}, \eta_2^{(0)})$.

This exponential family structure implies that, conditionally, the posterior factors are also in the same exponential families as the prior factors (Diaconis and Ylvisaker 1979),

$$p(\phi \mid y, \alpha) = h(\phi) \exp\left\{\eta(y, \alpha)^\top t(\phi) - a(y, \alpha)\right\},$$
$$p(\alpha_k \mid y_k, \phi) = h(\alpha_k) \exp\left\{\eta(y_k, \phi)^\top t(\alpha_k) - a(y_k, \phi)\right\}.$$

The global factor's natural parameter is $\eta(y, \alpha) = (\eta_1^{(0)} + \sum_{k=1}^{K} t(y_k, \alpha_k), \eta_2^{(0)} + \sum_{k=1}^{K} N_k)$.

With this statistical property at play—namely, that conjugacy gives rise to tractable conditional posterior factors—we can derive algorithms at a conditional level with exact iterative updates. This is assumed for most of the message passing of semiparametric models in Wand (2017). Importantly, this is not necessarily a limitation of the algorithm. It is a testament to leveraging model structure: without access to tractable conditional posteriors, additional approximations must be made. Wand (2017) provided an elegant way to separate out these nonconjugate pieces from the conjugate pieces.

In statistics, the most well-known example that leverages conditionally conjugate factors is the Gibbs sampling algorithm. From our own work, we apply the idea to access fast natural gradients in variational inference, which accounts for the information geometry of the parameter space (Hoffman et al. 2013). In other work, we demonstrate a collection of methods for gradient-based marginal optimization (Tran, Gelman, and Vehtari 2016). Assuming forms of conjugacy in the model class arrives at the classic idea of iteratively reweighted least squares as well as the EM algorithm. Such structure in the model provides efficient algorithms—both statistically and computationally—for their automated inference.

Open challenges and future directions. Message passing is a classic algorithm in the computer science literature, which is ripe with interesting ideas for statistical inference. In particular, MP enables new advancements in the realm of automated inference, where one can take advantage of statistical structure in the model. Wand (2017) made great steps following this direction.

With that said, important open challenges still exist to realize this fusion.

First is about the design and implementation of probabilistic programming languages. To implement Wand's (2017) message passing, the language must provide ways of identifying local structure in a probabilistic program. While that is enough to let practitioners use MP, a much larger challenge is to then automate the process of detecting local structure.

Second is about the design and implementation of inference engines. The inference must be extensible, so that users cannot only employ the algorithm in Wand (2017) but easily build on top of it. Further, its infrastructure must be able to encompass a variety of algorithms, so that users can incorporate MP as one of many tools in their toolbox.

Third, we think there are innovations to be made on taking the stance of modularity to a further extreme. In principle, one can compose not only localized message passing

updates but compose localized inference algorithms of any choice—whether it be exact inference, Monte Carlo, or variational methods. This modularity will enable new experimentation with inference hybrids and can bridge the gap among inference methods.

Finally, while we discuss MP in the context of automation, we point out that fully automatic algorithms are not possible. Associated with all inference are statistical and computational tradeoffs (Jordan 2013). Thus, we need algorithms along the frontier, where a user can explicitly define a computational budget and employ an algorithm achieving the best statistical properties within that budget; or conversely, define desired statistical properties and employ the fastest algorithm to achieve them. We think ideas in MP will also help in developing some of these algorithms.

References

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2016), "Stan: A Probabilistic Programming Language," *Journal of Statistical Software*, 76, 1–32. [156]

Diaconis, P., and Ylvisaker, D. (1979), "Conjugate Priors for Exponential Families," *The Annals of Statistics*, 7, 269–281. [157]

Gelman, A., Vehtari, A., Jylänki, P., Robert, C., Chopin, N., and Cunningham, J. P. (2014), "Expectation Propagation as a Way of Life," *arXiv preprint arXiv:1412.4869*. [157]

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013), "Stochastic Variational Inference," *Journal of Machine Learning Research*, 14, 1303–1347. [157]

Hoffman, M. D., and Gelman, A. (2014), "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," *Journal of Machine Learning Research*, 15, 1593–1623. [156]

Huang, Z., and Gelman, A. (2005), "Sampling for Bayesian Computation With Large Datasets," Technical Report. [157]

Jordan, M. I. (2013), "On Statistics, Computation and Scalability," *Bernoulli*, 19, 1378–1390. [158]

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2016), "Automatic Differentiation Variational Inference," *Journal of Machine Learning Research,* to appear. [156]

Tran, D., Gelman, A., and Vehtari, A. (2016), "Gradient-Based Marginal Optimization," Technical Report. [157]

Wand, M. P. (2017), "Fast Approximate Inference for Arbitrarily Large Semiparametric Regression Models via Message Passing," *Journal of the American Statistical Association*, this issue. [156,157]

Wang, X., and Dunson, D. B. (2013), "Parallelizing MCMC via Weierstrass Sampler," *arXiv preprint arXiv:1312.4605*. [157]

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2017, VOL. 112, NO. 517, Theory and Methods
http://dx.doi.org/10.1080/01621459.2016.1270045

Comment

Wanzhu Tu

Department of Biostatistics, Indiana University School of Medicine, Indianapolis, IN

1. Introduction

Matt Wand's article describes a computational paradigm for semiparametric regression through variational message passing (VMP), an approximate inference technique originated from Bayesian networks but found increasing application in mainstream statistical modeling. By expressing semiparametric models as factor graphs, Wand showed that factorized modeling components could be used as bases for fast estimation and inference, through mean field variational Bayes and VMP algorithms. I applaud Wand's effort in identifying and formulating common factor graph fragments for frequently used parametric and semiparametric models; these fragments are designed to function as the basic building blocks for more complex models and algorithms. Wand's work, in my opinion, represents an initial but important step toward the development of general-purpose model fitting algorithms and user-friendly software for larger semiparametric models.

Technical contributions aside, the article's dissemination of the computational details, illustrated by frequently used models together with real-data applications, makes it an excellent introduction to variational Bayes and related message passing algorithms in a more familiar modeling setting. The educational values of the piece should not be underestimated, as adoption of a new methodology is usually contingent on the pedagogical quality of its initial introduction. Wand deserves compliments for his clear articulation of the problem and explicit provision of the solution.

In this short discussion, I would like to recap some of the basic tenets of VMP, comment on its use in semiparametric regression, and give my reasons on why we should be excited about Wand's article, all from the viewpoint of someone who uses semiparametric methods in real scientific investigations. Of course, it would not be so interesting if I am in complete agreement with Wand—I will comment on a few things that I thought deserved greater attention.

2. VMP in a Nutshell

VMP is an approximate inference algorithm that initially arises in the context of graphical models. An earlier algorithm based on a similar idea appeared under the name of belief propagation (Pearl 1986; Spiegelhalter 1986; Lauritzen and Spiegelhalter 1988). The VMP technique as we currently understand is framed by Winn (2003) and Winn and Bishop (2005), upon which this review is based.

CONTACT Wanzhu Tu *wtu1@iu.edu* Department of Biostatistics, Indiana University School of Medicine, Indianapolis, IN 46202.

Variational inference, in its essence, is a process of approximation. Its basic idea is fairly straightforward: letting $\mathbf{D}$ and $\mathbf{H}$ be the observed and the unobserved variables, one approximates the true posterior distribution $P(\mathbf{H}|\mathbf{D})$ by a variational distribution $Q(\mathbf{H})$ that minimizes of $\mathrm{KL}(Q||P)$, the Kullback–Leibler divergence between the true posterior P and its approximate Q. Winn (2003) expressed the Kullback–Leibler divergence as

$$\mathrm{KL}(Q||P) = \int_{\mathbf{H}} Q(\mathbf{H}) \log \frac{Q(\mathbf{H})}{P(\mathbf{H}, \mathbf{D})} d\mathbf{H} + \log P(\mathbf{D}).$$

Since the last term in the above equation has no reliance on Q, it suffices to minimize the first term, or to maximize its negative, which is often referred to as the variational lower bound of Q, denoted as $\mathcal{L}(Q)$.

The burden of maximizing $\mathcal{L}(Q)$ depends on the structural complexity of $Q(\mathbf{H})$. Factorizing the hidden variables as $Q(\mathbf{H}) = \prod_i Q_i(\mathbf{H}_i)$ simplifies the calculation of Q; here, the Q_i's represent the variational distributions for the disjoint groups of variables. Winn and Bishop (2005) provided updating formulas that iteratively maximize the lower bound $\mathcal{L}(Q)$, by cycling through all factors.

The factorization here actually does more than just simplifying the structure of $Q(\mathbf{H})$. It helps to localize the computation, so that the updating equation for a node H_j only involves the variables relates to H_j, including its parents, children, and children's parents. This localized network is sometimes referred to as the Markov blanket of H_j. Winn and Bishop (2005) proposed to decompose the optimization into a series of local operations that depend only on "messages" passed among neighboring nodes. This sets the stage for VMP. Operationally, the messages from parents to children are simply the expectation of the sufficient statistics under Q, and the messages from children are the natural parameters, which happen to rely on messages previously received from co-parents. For nodes whose values are observed, expectations are replaced by the observed values.

An important feature of the standard VMP algorithm is its use of the conjugate-exponential model. When the conditional distribution of a node, given its parents, follows an exponential distribution and is conjugate with respect to the prior distributions of its parents, one has a conjugate-exponential model. Such a model ensures that all nodes and their respective priors are within the same family of distributions. As a result, one could simply focus on the parameters without worrying about the distribution functions. More specifically, the conjugate-exponential model provides sufficient statistic and natural parameter vectors in closed forms, which are used for message calculation and variational updating. By updating the natural parameter and sufficient statistic vectors for each node, one iteratively calculates and improves the lower bounds $\mathcal{L}(Q)$.

The use of conjugate-exponential models, however, is not just a trick for computational expedience. It ensures that optimization of the variational distribution is carried out within the distributional family of the true posterior. It provides an implicit but important assurance that the approximation has some functional resemblance to its target; this is something that unconstrained minimization of Kullback–Leibler divergence is unable to provide. For all practical purposes, this constraint is not more limiting than the distributional assumptions used by other analytical frameworks. When necessary, extensions to nonconjugate conditional distributions are possible, typically by approximating the nonconjugate distributions with conjugate ones.

3. VMP: The Knowns and Unknowns

Much has been learned in the last decade about the methodological properties and operational characteristics of VMP. First, VMP is known to be deterministic—under the same initialization and message passing schedule, the algorithm would lead to the same solution. Second, VMP algorithms always converge, because the stopping rules are based either on the improvement in the lower bound or on the stabilization of marginal distributions. No objective criteria on the quality of approximation are used. Therefore, algorithmic convergence does not guarantee an achievement of the right solution. Fortunately, numerical evidence from models tested so far has shown generally good agreement between the VMP and MCMC estimates. In semiparametric regression, Wand showed data examples where VMP and MCMC methods produced similar results. Finally, there is a recognition on the similarity between expectation–maximization (EM) and VMP algorithms. A key difference though is that EM uses posterior modes (or MLE) as point estimates of parameters of interest while estimating the posterior distributions for the latent variables. Variational Bayes, on the other hand, makes no distinction between model parameters and latent variables. In a factorized approximation, VMP simply cycles through the latent variables and model parameters, running localized message passing and variational updating.

We know considerably less about VMP's capacity for statistical inference. Winn and others have observed a tendency of VMP to concentrate probability mass in regions of high posterior probability while depleting the probability mass in regions of low posterior probability, for example, Figure 4 in Winn and Bishop (2005). Although this may not have negatively affected point estimation, inference could be put on a shakier ground. In Section 5.4 of his article, Wand observed that the pointwise 95% VMP credible intervals for the probit model were narrower than those produced by MCMC. The differences were not large in Wand's example, and they were attributed to the lack of proper accounting of correlations among the parameters. While this may be true in Wand's case, more concentrated probability mass has been observed in other situations as well, which could be a cause of concern. Could it be due to the behavior of Kullback–Leibler divergence measure that is used to derive the variational distribution? Regardless, as we move to apply VMP techniques to more complicated models, it seems reasonable to demand a more detailed understanding of VMP's capacity for valid inference in different modeling and data situations. Future investigations perhaps should place more emphasis on inference than on estimation.

4. VMP in Semiparametric Regression

Semiparametric regression, which combines the strengths of traditional parametric regression with enhanced modeling flexibility of nonparametric techniques, has emerged as one of the most vibrant and fast-developing branch of modern statistics. Besides its intrinsic methodological appeals, semiparametric

regression owes much of its success to the unified treatment of a diverse class of models and its mixed effect model representation, as elegantly described by Ruppert, Wand, and Carroll (2003). Such a representation allows fitting of most semiparametric regression models with standard computational software.

Semiparametric analysis in scientific investigation has led to important findings that would not have been made with traditional parametric methods. For example, in a series of mechanistic studies of blood pressure regulation, investigators have used semiparametric methods to examine the biological interactions among components of the renin–angiotensin–aldosterone system (Tu et al. 2012; Yu et al. 2013). A key observation is that individuals of African ancestry tend to have increased levels of sensitivity to aldosterone, a mineralocorticoid hormone that up-regulates sodium reabsorption by the kidney, and this increased sensitivity is compounded by an expansion of extracellular fluid volume (Tu et al. 2014). This discovery not only helps to explain why African Americans, under similar conditions, would be more susceptible to hypertension than whites, but also lead to a suggestion of a new therapeutic strategy—blocking aldosterone in patients who do not have particularly high levels of aldosterone in the circulation (Funder 2014). It may be difficult to imagine that all these findings originate from a series of semiparametric analyses of observational data. But the initial observations are confirmed by evidence from human experiments.

The above example points to the power of semiparametric analysis. But if semiparametric models can be implemented with standard software, who would need VMP? People in modeling practice are well aware of the limitations of existing computational tools. Again, in the renin–angiotensin–aldosterone system there are in fact multiple operators that jointly determine the levels of sodium retention and thus blood pressure. These biological influences are often intertwined and nonlinear. A more realistic depiction of the physiologic system would result in very complex semiparametric models or model systems. Previously, we have attempted to develop multivariate semiparametric models that accommodate some aspects of the interactive features, in a much simpler situation (Liu and Tu 2012). Although those models are nowhere near the level of sophistication needed to accurately convey the current understanding of human physiology, they are at the limit of our computational capacity. As many modelers would confess, it is usually not so difficult to write a good model—the challenge is to fit it. Considering the factors involved and the interactive relations among them, fitting a real-mechanism-driven model using existing tools has so far been a hopeless exercise. This is precisely where VMP might come to the rescue, and why Wand's work is so exciting.

5. Where Do We Go From Here?

Over the past decade, VMP has grown and matured into an accepted computational technique and inference tool. Its success in a wide variety of statistical models has attested to its value. Its most recent penetration into the field of semiparametric regression brings about new excitement and expectation. By providing closed-form VMP solutions for a broad class of semiparametric models, Wand's article brings us one step closer to acquiring the necessary computational tools for larger and more complex semiparametric models.

This, of course, is not to say that the algorithms that the article presented are ready for off-the-shelf use in practical data analysis. We are not quite there yet, and I do not expect the implementation will be trivial. In my mind, Wand's article is not so much about programming and software development, but more about a model-fitting methodology. The VMP building blocks that Wand presented are indeed implementable. But it is unclear how likely practical data analysts would go out of their ways to program the algorithms, just for the purpose of fitting standard semiparametric models. The true and bigger impact of Wand's work is likely to be on modelers. These algorithms and their basic building blocks will give modelers a freer hand in constructing new semiparametric models that better answer real scientific questions.

A number of other issues may require further investigation. One is VMP's performance in statistical inference, in comparison with MCMC and penalized likelihood methods. Statistical inference in semiparametric models has its unique challenges, but inference is what science demands. A proven ability to produce valid inference is always expected of a new method. Another issue requires more investigation is VMP's balance between computational efficiency and quality of approximation in semiparametric regression, again in comparison with stochastic and likelihood based methods. A better understanding of these issues will help analysts make more informed decisions on modeling approaches.

Will VMP impact science in ways that other major methodological advancements have in the past? Only time will tell. Things will have to be worked out on a case-by-case basis, but we have every reason to be hopeful. Stay tuned.

Funding

The author's work is supported by NIH grants HL095086 and AA025208. The views expressed by the author in this commentary are his own.

References

Funder, J. W. (2014), "Sensitivity to Aldosterone: Plasma Levels are Not the Full Story," *Hypertension*, 63, 1168–1170. [160]

Lauritzen, S. L., and Spiegelhalter, D. J. (1988), "Local Computations With Probabilities on Graphical Structures and Their Application to Expert Systems," *Journal of the Royal Statistical Society*, 50, 157–224. [158]

Liu, H., and Tu, W. (2012), "A Semiparametric Regression Model for Paired Longitudinal Outcomes With Application in Childhood Blood Pressure Development," *Annals of Applied Statistics*, 6, 1861–1882. [160]

Pearl, J. (1986), "Fusion, Propagation and Structuring in Belief Networks," *Artificial Intelligence*, 29, 241–288. [158]

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, Cambridge, UK: Cambridge University Press. [160]

Spiegelhalter, D. J. (1986), "Probabilistic Reasoning in Predictive Expert Systems," in *Uncertainty in Artificial Intelligence*, eds. L. N. Kanal, and J. F. Lemmer, Amsterdam, The Netherlands: North Holland, pp. 47–68. [158]

Tu, W., Eckert, G. J., Pratt, J. H., and Danser, A. J. (2012), "Plasma Levels of Prorenin and Renin in Blacks and Whites: Their Relative Abundance and Associations With Plasma Aldosterone Concentration," *American Journal of Hypertension*, 25, 1030–1034. [160]

Tu, W., Eckert, G. J., Hannon, T. S, Liu, H., Pratt, L. M., Wagner, M. A, DiMeglio, L. A., Jung, J., and Pratt, J. H. (2014), "Racial Differences

in Sensitivity of Blood Pressure to Aldosterone," *Hypertension*, 63, 1212–1218. [160]
Winn, J. (2003), "Variational Message Passing and Its Application," PhD thesis, University of Cambridge. [158,159]
Winn, J., and Bishop, C. M. (2005), "Variational Message Passing," *Journal of Machine Learning Research*, 6, 661–694. [158,159]
Yu, Z., Eckert, G. J., Liu, H., Pratt, J. H., and Tu, W. (2013), "Adiposity has Unique Influence on the Renin-Aldosterone Axis and Blood Pressure in Black Children," *Journal of Pediatrics*, 163, 1317–1322. [160]

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2017, VOL. 112, NO. 517, Theory and Methods
http://dx.doi.org/10.1080/01621459.2016.1270049

Comment

Philip T. Reiss[a,b] and Jeff Goldsmith[c]

[a]Department of Statistics, University of Haifa, Haifa, Israel; [b]The Department of Child and Adolescent Psychiatry, New York University School of Medicine, New York, NY; [c]Department of Biostatistics, Columbia University Mailman School of Public Health, New York, NY

1. Introduction

This very stimulating article reminded us of the following remark in a review article by Wand and co-authors (Ruppert, Wand, and Carroll 2009):

> Interplay with Computer Science is one of the most exciting recent developments in semiparametric regression. We anticipate this to be an increasingly fruitful area of research.

Bringing message passing to bear on semiparametric regression, as Wand has done here, is very much in the spirit of such interplay. The notion of message passing is ubiquitous in some areas of computer science, such as distributed computing and object-oriented programming. More specifically, within the field of artificial intelligence, the influential problem-solving model of Hewitt (1977) is based upon message passing, while Pearl (1982, 1988) proposed the passing of messages among neighboring nodes as a way to update beliefs efficiently in large Bayesian networks.

Against this backdrop it is unsurprising that, whereas variational message passing (VMP) is formulated quite differently in the three papers that Wand cites (Winn and Bishop 2005; Minka 2005; Minka and Winn 2008), in each case the authors find it natural to portray the algorithm as passing messages among the nodes of a network. But for readers like us with a mainstream statistics background, a message passing scheme such as that described by Wand comes across, at least at first, as uncomfortably mysterious (see Wainwright and Jordan 2008, p. 36).

We begin by posing three questions that arose for us as we read Wand's article. The first crystallizes our unease with the very notion of message passing, and addressing this key question will pave the way toward answering the other two.

1. As presented by Wand, following Minka (2005), VMP works by iteratively updating two types of messages: messages $m_{\boldsymbol{\theta}_i \to f_j}(\boldsymbol{\theta}_i)$ from variables (stochastic nodes) to factors, and messages $m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)$ from factors to variables. What, exactly, is the statistical meaning of these messages?
2. How is the VMP algorithm related to the traditional approach to mean field variational Bayes (MFVB)?
3. Wand's message updates are given in (W7)–(W9) (here and below, to avoid confusion with our own equation numbers, we use (Wx) to denote Wand's equation (x)). How do these reduce to natural parameter updates, as presented from Section 3.2 onward?

In the following sections, we attempt to answer these questions and thereby, we hope, to shed some light on VMP.

2. A Closer Look at Messages

To address Question 1, we consider first the variable-to-factor messages and then the factor-to-variable messages.

Variable-to-factor messages. Recall that the form of the messages in Wand's presentation of VMP flows from the factor graph representation. In an article popularizing this representation, Kschischang, Frey, and Loeliger (2001) developed a generic *sum–product* algorithm in which messages are passed back and forth between factors and variables, as in Wand's presentation of VMP. Bishop (2006, p. 408) noted that one can eliminate the variable-to-factor messages in the sum–product algorithm, and reformulate it with only factor-to-variable messages. We find it helpful to reformulate VMP in a similar way.

Let us first recall Wand's generic algorithm. In Section 3.2, following Minka (2005), he presents an iteration loop for VMP that could be stated as (1) choose a factor; (2) update messages from neighboring stochastic nodes to that factor; (3) update messages from that factor to neighboring stochastic nodes. While the schedule for updating factors may be flexible in some applications (Winn and Bishop 2005, sec. 3.5), for our purposes we can assume the factors are updated serially in a fixed order. Thus, a single iteration of the VMP algorithm might be written as a loop over j, with each step comprising two subloops:

Loop A. For $j = 1, \ldots, N$:

CONTACT Philip T. Reiss *reiss@stat.haifa.ac.il* Department of Statistics, University of Haifa, Haifa 31905, Israel.

1. For each $i' \in S_j$, perform the update

$$m_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'}) \longleftarrow\propto \prod_{j' \neq j : i' \in S_{j'}} m_{f_{j'} \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'}). \tag{1}$$

This is just (W7), but with $S_{j'}$ (defined in (W5)) replacing the equivalent "neighbors(j')."

2. For each $i \in S_j$:
   (a) Define the density in (W9), which is proportional to

$$\prod_{i' \in S_j \setminus \{i\}} m_{f_j \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'}) m_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'}). \tag{2}$$

   (b) Update the factor-to-variable message $m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)$ by (W8), which we repeat for convenience, again using S_j in place of "neighbors(j)":

$$m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \longleftarrow\propto \exp\left[E_{f_j \to \boldsymbol{\theta}_i}\left\{\log f_j(\boldsymbol{\theta}_{S_j})\right\}\right], \tag{3}$$

   where the expectation is with respect to the density in (2).

The messages on the right-hand side of (1) emanate from factors other than f_j, and thus are not updated within the current step of the loop over j. Therefore, the density (2) is unchanged if we substitute the right-hand side of (1) for $m_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'})$ in (2). Doing so renders the first subloop redundant, so that a single iteration of VMP can be rewritten in the following mathematically equivalent form.

Loop B. For $j = 1, \ldots, N$:
For each $i \in S_j$,
(a) Define the density proportional to

$$\prod_{i' \in S_j \setminus \{i\}} \left[m_{f_j \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'}) \prod_{j' \neq j : i' \in S_{j'}} m_{f_{j'} \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'}) \right] = \prod_{i' \in S_j \setminus \{i\}} \prod_{j' : i' \in S_{j'}} m_{f_{j'} \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'}). \tag{4}$$

(b) Update the factor-to-variable message $m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)$ using (3), with the expectation taken with respect to the density in (4).

If the jth factor depends on more than two of the $\boldsymbol{\theta}_i$ (i.e., $|S_j| > 2$) then Loop A may save some computation by performing the multiplication (1) just once, whereas Loop B must do the same multiplication, in (4), $|S_j| - 1$ times. We suspect the savings would typically be small, since $|S_j| \leq 2$ for most j and the multiplication reduces to the summing natural parameters (see Section 4). At any rate, the conceptual simplicity that Loop B achieves by doing away with variable-to-factor messages will facilitate our development in Sections 3 and 4.

Factor-to-variable messages. As Wand notes, in MFVB we seek component densities $q_1^*(\boldsymbol{\theta}_1), \ldots, q_N^*(\boldsymbol{\theta}_N)$ (here and below, unlike Wand, we include subscripts for these densities) such that $q^*(\boldsymbol{\theta}) = \prod_{i=1}^M q_i^*(\boldsymbol{\theta}_i)$ minimizes the Kullback–Leibler divergence $\mathrm{KL}[q \| p(\cdot|\boldsymbol{D})] = \int q(\boldsymbol{\theta}) \log\left[\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{D})}\right] d\boldsymbol{\theta}$ over all product densities $q(\boldsymbol{\theta}) = \prod_{i=1}^M q_i(\boldsymbol{\theta}_i)$. By (W10), in the VMP implementation of MFVB, we have

$$q_i^*(\boldsymbol{\theta}_i) \propto \prod_{j : i \in S_j} m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \tag{5}$$

upon convergence. Alternatively, one can view $q_1^*(\boldsymbol{\theta}_1), \ldots, q_M^*(\boldsymbol{\theta}_M)$ as quantities that are being updated throughout the iterative algorithm (Minka 2005 does this). We can then view the factor-to-variable messages as (proportional to) iteratively updated subcomponent densities, where component $q_i^*(\boldsymbol{\theta}_i)$ is divided into subcomponents for each factor f_j of which $\boldsymbol{\theta}_i$ is an argument.

To summarize: we understand the variable-to-factor messages as a bookkeeping device with no independent statistical meaning, such that VMP can be formulated without them; and we interpret the factor-to-variable messages as factor-specific subcomponents of each component density as in (5). One could, then, jettison the message passing metaphor altogether, and replace the notation $m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)$ by $q_i^{f_j}(\boldsymbol{\theta}_i)$ or $q_i^j(\boldsymbol{\theta}_i)$.

3. Relating VMP to Traditional MFVB

We can now answer Question 2 posed in the Introduction. As explained in, for example, Ormerod and Wand (2010) and Goldsmith, Wand, and Crainiceanu (2011), the traditional MFVB algorithm initializes $q_1^*(\boldsymbol{\theta}_1), \ldots, q_M^*(\boldsymbol{\theta}_M)$ and then updates these iteratively via coordinate descent steps

$$q_i^*(\boldsymbol{\theta}_i) \longleftarrow\propto \exp\left[\int \log\{p(\boldsymbol{\theta}, \boldsymbol{D})\} \prod_{i' \neq i} \{q_{i'}^*(\boldsymbol{\theta}_{i'}) d\boldsymbol{\theta}_{i'}\}\right] \tag{6}$$

for $i = 1, \ldots, M$.
By (5), (4) reduces to $\prod_{i' \in S_j \setminus \{i\}} q_{i'}^*(\boldsymbol{\theta}_{i'})$, and thus the VMP update (3) can be rewritten as

$$m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \longleftarrow\propto \exp\left[\int \log\{f_j(\boldsymbol{\theta}_{S_j})\} \prod_{i' \in S_j \setminus \{i\}} \{q_{i'}^*(\boldsymbol{\theta}_{i'}) d\boldsymbol{\theta}_{i'}\}\right]. \tag{7}$$

This is a factor-specific analogue of the usual MFVB update (6): the ith component $q_i^*(\boldsymbol{\theta}_i)$ is replaced by its jth-factor-specific subcomponent $m_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)$, the joint density $p(\boldsymbol{\theta}, \boldsymbol{D})$ by its jth factor $f_j(\boldsymbol{\theta}_{S_j})$, and the product of $q_{i'}^*(\boldsymbol{\theta}_{i'}) d\boldsymbol{\theta}_{i'}$ over all $i' \neq i$ by one restricted to those in S_j (this last is not a real difference, since taking the product over all $i' \neq i$ in (7) would be equivalent).

The traditional MFVB algorithm cycles over all i (the variables) to update the component densities. VMP, on the other hand, cycles over j (the factors), and within each factor, cycles over $i \in S_j$ to update subcomponent densities.

4. Reduction to Natural Parameter Updates

We now turn to Question 3. Throughout his Sections 3 and 4, Wand exploits conjugacy (as defined for factor graphs in Section 3.2.2) to simplify a number of special cases of the VMP algorithm, and in particular, to reduce updates (W7)–(W9) for the *messages* to updates for the *natural parameters* of the messages. The following is our attempt to make explicit what Wand's treatment assumes implicitly.

In the exponential family case, it is natural to write the factor-to-variable messages in the form

$$m_{f_j} \to \boldsymbol{\theta}_i(\boldsymbol{\theta}_i) \propto \exp\left[T_i(\boldsymbol{\theta}_i)^T \boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i}\right] \tag{8}$$

for a natural parameter $\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}$. But since these messages are not really defined—only their updates are, by (3)—there is a hint of vagueness in the definition of $\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}$. It seems to us that the key to avoiding such ambiguity is to start by defining the jth factor density as an exponential family density respect to $\boldsymbol{\theta}_i$, for each $i \in S_j$: namely

$$f_j(\boldsymbol{\theta}_{S_j}) \propto \exp\left[T_i(\boldsymbol{\theta}_i)^T \boldsymbol{\eta}_{f_j\setminus\boldsymbol{\theta}_i}\right], \tag{9}$$

where $\boldsymbol{\eta}_{f_j\setminus\boldsymbol{\theta}_i}$ does not depend on $\boldsymbol{\theta}_i$. For instance, consider the factor $f_j(\sigma^2, a) = p(\sigma^2|a)$ in Wand's linear regression example. In Section S.2.1 of the supplement, the logarithm of this factor is written in the two forms:

$$\log p(\sigma^2|a) = T(\sigma^2)^T \boldsymbol{\eta}_{p(\sigma^2|a)\setminus\sigma^2} = T(a)^T \boldsymbol{\eta}_{p(\sigma^2|a)\setminus a}$$

where $T(\cdot)$ denotes the inverse chi-squared sufficient statistic vector,

$$\boldsymbol{\eta}_{p(\sigma^2|a)\setminus\sigma^2} = \begin{bmatrix} -3/2 \\ -1/(2a) \end{bmatrix}, \text{ and } \boldsymbol{\eta}_{p(\sigma^2|a)\setminus a} = \begin{bmatrix} -1/2 \\ -1/(2\sigma^2) \end{bmatrix}. \tag{10}$$

Inserting (9) into (the log of) (3) yields

$$\begin{aligned} \log m_{f_j\to\boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) &\longleftarrow E_{f_j\to\boldsymbol{\theta}_i}\left[\log f_j(\boldsymbol{\theta}_{S_j})\right] + \text{constant} \\ &= T_i(\boldsymbol{\theta}_i)^T E_{f_j\to\boldsymbol{\theta}_i}(\boldsymbol{\eta}_{f_j\setminus\theta_i}) + \text{constant}. \end{aligned}$$

This update equation serves as the justification both for writing factor-to-variable messages in the form (8) and for updating their natural parameters by

$$\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i} \longleftarrow E_{f_j\to\boldsymbol{\theta}_i}(\boldsymbol{\eta}_{f_j\setminus\boldsymbol{\theta}_i}). \tag{11}$$

The density in Loop A is proportional to (2), which can be expressed as

$$\begin{aligned} &\prod_{i'\in S_j\setminus\{i\}} \exp\left[T_{i'}(\boldsymbol{\theta}_{i'})^T(\boldsymbol{\eta}_{f_j\to\theta_{i'}} + \boldsymbol{\eta}_{\theta_{i'}\to f_j})\right] \\ &= \prod_{i'\in S_j\setminus\{i\}} \exp\left[T_{i'}(\boldsymbol{\theta}_{i'})^T \boldsymbol{\eta}_{f_j\leftrightarrow\theta_{i'}}\right], \end{aligned} \tag{12}$$

with $\boldsymbol{\eta}_{f_j\leftrightarrow\theta_{i'}}$ generalizing the notation that first appears in (W22). Eliminating variable-to-factor messages as in Loop B above, (12) becomes

$$\prod_{i'\in S_j\setminus\{i\}} \exp\left[T_{i'}(\boldsymbol{\theta}_{i'})^T \sum_{j':i'\in S_{j'}} \boldsymbol{\eta}_{f_{j'}\to\theta_{i'}}\right] \tag{13}$$

(cf. (4)). If, as suggested in Section 3 above, $q^*_{i'}(\boldsymbol{\theta}_{i'})$ is defined using (5) throughout the iterations rather than just at convergence, then (13) can be rewritten as

$$\prod_{i'\in S_j\setminus\{i\}} \exp[T_{i'}(\boldsymbol{\theta}_{i'})^T \boldsymbol{\eta}_{q^*_{i'}(\theta_{i'})}].$$

This obviates the need for the notation $\boldsymbol{\eta}_{f_j\leftrightarrow\theta_{i'}}$, which is equivalent to $\boldsymbol{\eta}_{q^*_{i'}(\theta_{i'})}$.

We can thus summarize the natural-parameter-updating version of VMP as follows. First, we must obtain the right side of (11), a function of $\{\boldsymbol{\eta}_{q^*_{i'}(\theta_{i'})} : i' \in S_j\setminus\{i\}\}$, for each i, j. Then, after initializing all the natural parameters $\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}$, each iteration of the algorithm is a modified, more concrete version of Loop B:

Loop C. For $j = 1, \ldots, N$:
For each $i \in S_j$,
(a) Compute $\boldsymbol{\eta}_{q^*_{i'}(\theta_{i'})} = \sum_{j':i'\in S_{j'}} \boldsymbol{\eta}_{f_{j'}\to\theta_{i'}}$ for each $i' \in S_j\setminus\{i\}$;
(b) Plug these into the right side of (11) to update $\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}$.

Table 1. Factor-to-variable message natural parameters for the linear regression example of Wand's Section 3.

	$p(\boldsymbol{\beta})$	$p(\boldsymbol{y}\|\boldsymbol{\beta},\sigma^2)$	$p(\sigma^2\|a)$	$p(a)$	
$\boldsymbol{\beta}$	$\boldsymbol{\eta}_{p(\beta)\to\beta}$	$\boldsymbol{\eta}_{p(y\|\beta,\sigma^2)\to\beta}$			$\boldsymbol{\eta}_{q^*(\beta)}$
σ^2		$\boldsymbol{\eta}_{p(y\|\beta,\sigma^2)\to\sigma^2}$	$\boldsymbol{\eta}_{p(\sigma^2\|a)\to\sigma^2}$		$\boldsymbol{\eta}_{q^*(\sigma^2)}$
a			$\boldsymbol{\eta}_{p(\sigma^2\|a)\to a}$	$\boldsymbol{\eta}_{p(a)\to a}$	$\boldsymbol{\eta}_{q^*(a)}$

5. Bayesian Linear Regression Revisited

The remarkably simple Loop C can be visualized using a less snazzy alternative to a factor graph: an $N \times M$ table of natural parameters for the factor-to-variable messages, with a row for each of the variables $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ and a column for each of the factors $f_1, \ldots, f_M$. Table 1 illustrates this for the linear regression example of Wand's Section 3 (here, like Wand, we dispense with the subscripts in q^*_i).

In this case, only the second and third columns require updates (see (W21)), so each iteration consists of updating these two columns in turn. Consider updating the third column, that is, the natural parameters for the messages from $p(\sigma^2|a)$ to σ^2 and a. By (11), these updates are expectations of (10) with respect to densities proportional to (13); formulas for these expectations are as given in (W24), using information in Wand's Table S.1. Given these update formulas, in Loop C we simply add the natural parameters in the third row of Table 1 and insert the result ($\boldsymbol{\eta}_{q^*(a)}$, or $\boldsymbol{\eta}_{p(\sigma^2|a)\leftrightarrow a}$ in Wand's notation) into the formula for $\boldsymbol{\eta}_{p(\sigma^2|a)\to\sigma^2}$ to update this natural parameter; and then, similarly, we add up the natural parameters in the second row of Table 1 and insert the result into the formula for updating $\boldsymbol{\eta}_{p(\sigma^2|a)\to a}$.

6. Summary and Conclusion

In this comment, we have sought to build on the pedagogical component of Wand's achievement: that is, rendering VMP intelligible to statisticians who, like us and unlike Wand, have *not* spent years painstakingly reformulating and extending the methodology. Our main proposal, Loop B, is a modified VMP algorithm that is equivalent to the original, mathematically if not computationally, but does away with the variable-to-factor messages. This modification offers a reduced notational load, a clearer connection to the traditional implementation of MFVB, and a streamlined account of the reduction from message updates to natural parameter updates.

Of course, while explaining VMP to statisticians is an important contribution in itself, Wand has gone a great deal further. He has masterfully laid the groundwork for VMP-based semiparametric regression, which should be a huge step forward for flexible modeling with large datasets. Good on him (as they say Down Under) for this major contribution. We look forward to

further advances in this area by Wand and coworkers, and by others who will draw inspiration from this landmark article.

Funding

Reiss's research was supported by award R01 MH095836 from the National Institute of Mental Health, and Goldsmith's research was supported in part by awards R01HL123407 from the National Heart, Lung, and Blood Institute and R21EB018917 from the National Institute of Biomedical Imaging and Bioengineering.

References

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, New York: Springer. [161]

Goldsmith, J., Wand, M. P., and Crainiceanu, C. (2011), "Functional Regression via Variational Bayes," *Electronic Journal of Statistics*, 5, 572–602. [162]

Hewitt, C. (1977), "Viewing Control Structures as Patterns of Passing Messages," *Artificial Intelligence*, 8, 323–364. [161]

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001), "Factor Graphs and the Sum–Product Algorithm," *IEEE Transactions on Information Theory*, 47, 498–519. [161]

Minka, T. (2005), "Divergence Measures and Message Passing," Technical Report MSR-TR-2005-173, Microsoft Research. [161,162]

Minka, T., and Winn, J. (2008), "Gates: A Graphical Notation for Mixture Models," Technical Report MSR-TR-2008-185, Microsoft Research. [161]

Ormerod, J. T., and Wand, M. P. (2010), "Explaining Variational Approximations," *The American Statistician*, 64, 140–153. [162]

Pearl, J. (1982), "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach," In *AAAI-82 Proceedings*, pp. 133–136. [161]

——— (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo, CA: Morgan Kaufmann. [161]

Ruppert, D., Wand, M. P., and Carroll, R. J. (2009), "Semiparametric Regression During 2003–2007," *Electronic Journal of Statistics*, 3, 1193–1256. [161]

Wainwright, M. J., and Jordan, M. I. (2008), "Graphical Models, Exponential Families, and Variational Inference," *Foundations and Trends in Machine Learning*, 1, 1–305. [161]

Winn, J., and Bishop, C. M. (2005), "Variational Message Passing," *Journal of Machine Learning Research*, 6, 661–694. [161]

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2017, VOL. 112, NO. 517, Theory and Methods
http://dx.doi.org/10.1080/01621459.2016.1270050

Comment

Simon N. Wood

School of Mathematics, University of Bristol, Bristol, United Kingdom

Matt Wand's article represents a very substantial contribution to the methods available for Bayesian inference with semiparametric models. The implicit focus on models for grouped data with large numbers of group and/or subject specific smooth effects is particularly welcome, and nicely complements the major alternative simulation-free approach to fully Bayesian inference with large models of Rue, Martino, and Chopin (2009, INLA), where the focus is on models that are large because of high-rank spatial fields. INLA and VMP share the property of being computationally feasible for much bigger problem sizes than are readily tackled using stochastic simulation, at the expense of approximations that are typically very accurate. For both approaches, efficient computation rests on the exploitation of model sparsity, as is generally the case for methods for inference with large models.

Indeed if we recognize the centrality of sparsity then a very simple approach to approximate inference with some very large semiparametric models is possible. In particular, consider an empirical Bayes approach using sparse matrix methods (e.g., Davis 2006) and the generalized Fellner–Schall smoothing parameter estimation algorithm of Wood and Fasiolo (2017). A concrete example is the following regression model:

$$y_i = \alpha + f_1(x_{1i}) + f_2(x_{2i}) + f_{a(i)}(z_{1i}) + f_{b(i)}(z_{2i}) + f_{s(i)}(z_{3i}) + \epsilon_i,$$

where the $f_\cdot$ are all smooth functions, the x_j and z_j are covariates, and $s(i)$ indicates the subject to which the ith observation belongs. Furthermore, with each subject is associated one level of each of the two crossed factors a and b: $a(i)$ and $b(i)$ indicate the levels of a and b associated with observation i. In the example below we assume that there are 10,000 subjects and that a and b have, respectively, 30 and 20 levels. Each subject has measurements at 90 values of the covariates, and the smooth functions are each represented with rank 10 spline bases. So the model has 10,052 smooth functions, 100,521 coefficients, and there are 900,000 observations. Five smoothing parameters were used, one each for f_1 and f_2, one for the 30 f_a smooths and so on. For comparison, this model has around 10 times as many coefficients and 10 times as many data as the models referred to in sec. 6, from Lee and Wand (2016).

We can write the model as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ and estimate the coefficients to minimize the penalized least-square objective $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_j \boldsymbol{\beta}^\mathsf{T}\mathbf{S}_\lambda\boldsymbol{\beta}$, where $\mathbf{S}_\lambda = \sum_j \lambda_j \mathbf{S}_j$. Formally the penalized least-square estimates are then $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$. If $\mathbf{X}$ were a dense matrix, the computations would be infeasible, but for examples like the one given $\mathbf{X}$ and $\mathbf{X}^\mathsf{T}\mathbf{X} + \mathbf{S}_\lambda$ are in fact highly sparse. This means that we can solve for $\hat{\boldsymbol{\beta}}$ using the sparse Cholesky decomposition $\mathbf{L}^\mathsf{T}\mathbf{L} = \mathbf{P}(\mathbf{X}^\mathsf{T}\mathbf{X} + \mathbf{S}_\lambda)\mathbf{P}^\mathsf{T}$, where $\mathbf{P}$ is a pivoting matrix, designed

CONTACT Simon N. Wood simon.wood@bath.edu School of Mathematics, University of Bristol, Tyndall Ave, Bristol BS8 1TH, United Kingdom.

to ensure that $\mathbf{L}$ is sparse (for efficiency reasons the pivoting matrix is never formed explicitly). Then $\hat{\boldsymbol{\beta}} = \mathbf{P}^{\mathsf{T}}\mathbf{L}^{-1}\mathbf{L}^{-\mathsf{T}}\mathbf{P}\mathbf{X}^{\mathsf{T}}\mathbf{y}$, that is, a computation involving two sparse triangular solves and two pivoting operations (see Davis 2006, for details).

The extended Fellner–Schall algorithm of Wood and Fasiolo (2016) provides a way of iteratively updating the smoothing parameters to maximize the restricted marginal likelihood of the model based on an explicit formula, which, crucially, maintains sparsity of the computations. Updates are possible for models with any Fisher regular likelihood, but in the current context the update is

$$\lambda_j^* = \sigma^2 \frac{\mathrm{tr}(\mathbf{S}_\lambda^- \mathbf{S}_j) - \mathrm{tr}\{(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\mathbf{S}_j\}}{\hat{\boldsymbol{\beta}}^{\mathsf{T}}\mathbf{S}_j\hat{\boldsymbol{\beta}}}\lambda_j.$$

$\mathbf{S}_\lambda^-$ is a pseudoinverse of $\mathbf{S}_\lambda$, and for a sparse block diagonal $\mathbf{S}_\lambda$ (as in the example model), $\mathrm{tr}(\mathbf{S}_\lambda^- \mathbf{S}_j)$ is efficiently computable. In fact for the example model itself, where no term has more than one associated smoothing parameter, $\mathrm{tr}(\mathbf{S}_\lambda^- \mathbf{S}_j) = \mathrm{rank}(\mathbf{S}_j)/\lambda_j$. The term $\mathrm{tr}\{(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\mathbf{S}_j\}$ is the sum of squares of the elements of the matrix $\mathbf{B}$ where $\mathbf{B} = \mathbf{L}^{-\mathsf{T}}\mathbf{P}\mathbf{D}_j$ (i.e., the result of a sparse triangular solve and a permutation), and $\mathbf{D}_j$ is any sparse matrix such that $\mathbf{D}_j\mathbf{D}_j^{\mathsf{T}} = \mathbf{S}_j$. $\mathbf{D}_j$ can readily be created alongside $\mathbf{S}_j$. Note that $\mathrm{tr}\{(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{X}\} = p - \sum_j \lambda_j \mathrm{tr}\{(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\mathbf{S}_j\}$ gives the model effective degrees of freedom, which is useful for estimating σ^2. To estimate the whole model requires iteratively computing $\hat{\boldsymbol{\beta}}$ given $\hat{\boldsymbol{\lambda}}$, and then updating $\hat{\boldsymbol{\lambda}}$ given the current $\hat{\boldsymbol{\beta}}$. The results are illustrated in Figure 1. For the example, model setup and estimation took less than 5 min on an old mid-range laptop (i7-3540M 3GHz), with a memory footprint of less than 3Gb. Setup used package `mgcv` and computation the `Matrix` package (both recommended packages in R, R Core Team 2014).

Notice that the method generalizes immediately to large semiparametric GLMs: we simply require the usual GLM diagonal iteratively updated weight matrix so that $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ becomes $\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X}$, while $\mathbf{y}$ is replaced by the usual weighted iteratively updated pseudodata. In fact, generalizing to any regular likelihood that can be estimated by iteratively reweighted least squares is possible, as is further extension to models such as the GAMLSS class of Rigby and Stasinopoulos (2005). Note also that the method can deal with any smooth or random effect with a linear basis expansion where the penalty/precision matrix is linear in the smoothing/variance parameters.

The standard Bayesian interpretation of smoothing implies that $\boldsymbol{\beta}|\mathbf{y} \sim N(\hat{\boldsymbol{\beta}}, (\mathbf{X}^{\mathsf{T}}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\sigma^2)$, but $(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \mathbf{S}_\lambda)^{-1} = \mathbf{P}^{\mathsf{T}}\mathbf{L}^{-1}\mathbf{L}^{-\mathsf{T}}\mathbf{P}$, and $\mathbf{L}^{-\mathsf{T}}$ is sparse and cheap to compute. This sparsity is not inherited by $\mathbf{L}^{-1}\mathbf{L}^{-\mathsf{T}}$, but that is no bar to computing the marginal posterior distribution of any quantity dependent on $\boldsymbol{\beta}$. In general, if $\mathbf{c} = \mathbf{C}\boldsymbol{\beta}$, then the variances of the elements of $\mathbf{c}$ are the sums of squares of the rows of $\mathbf{C}\mathbf{P}^{\mathsf{T}}\mathbf{L}^{-1}$: this covers contrast curves, like those given in Figure 7 of Matt Wand's article, for example. The restriction to *marginal* posterior inference is shared with INLA, but not by VMP. Beyond the Gaussian case, the posterior for the method suggested here is a large sample approximation and will typically be less accurate than INLA or VMP, especially when the number of observations per group is small. Similarly such a simple empirical Bayes method cannot capture smoothing parameter uncertainty, nor deal with the more complex hierarchical model structures that VMP can tackle.

Generally then, sparsity will surely be the key to semiparametric big model inference in the future. And in particular, the potential for speed, accuracy, and generality that the VMP approach offers mean that there are good reasons to look forward with excitement to the arrival of the general purpose software that will surely follow Matt Wand's foundational article.

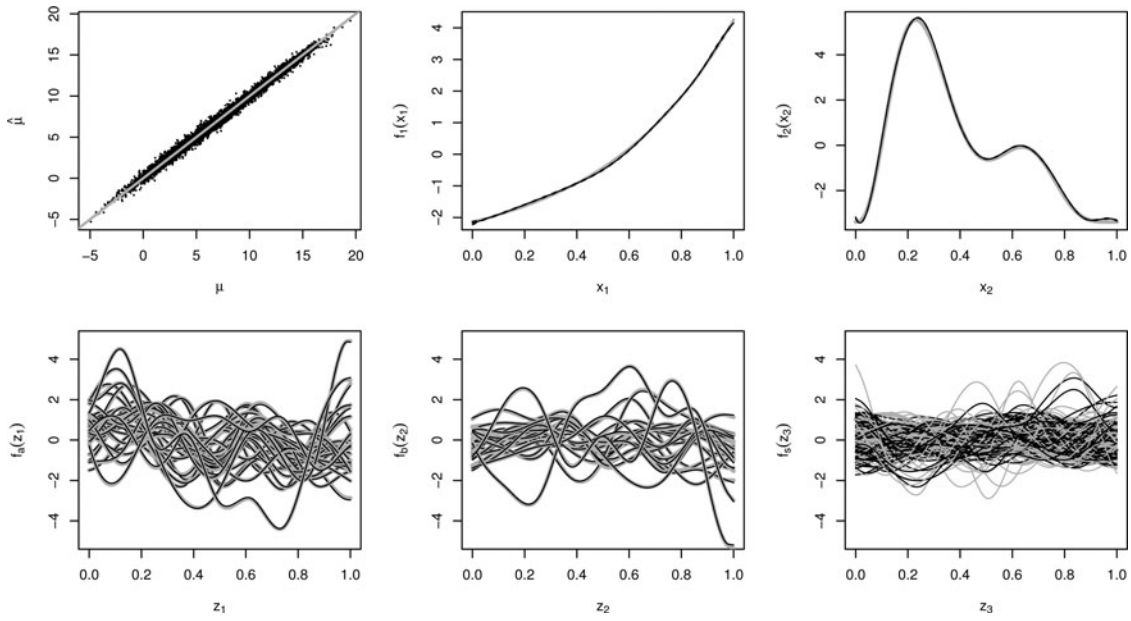


Figure 1. Results from fitting the 10^5 coefficient semiparametric model example from the text to 0.9M simulated data. The fit took less than 5 min and used less than 3Gb of memory on a rather modest laptop computer. Top left shows a 1% random sample of fitted values against simulation truth. The middle and right panels show the true f_1 and f_2 (gray) plus their estimates (black) with confidence intervals (dashed). The lower leftmost panel shows the 30 true functions and their estimates for the levels of crossed factor *a*. The middle lower panel is the same for crossed factor *b*. The lower right panel shows a 1% random sample of the subject-specific curves and their estimates. For this simulation, the error standard deviation was 1.5.

References

Davis, T. A. (2006), *Direct Methods for Sparse Linear Systems*, Philadelphia, PA: SIAM. [164]

Lee, C. Y. Y., and Wand, M. P. (2016), "Streamlined Mean Field Variational Bayes for Longitudinal and Multilevel Data Analysis," *Biometrical Journal*, 58, 868–895. [164]

R Core Team (2014), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing. [165]

Rigby, R., and Stasinopoulos, D. M. (2005), "Generalized Additive Models for Location, Scale and Shape," *Journal of the Royal Statistical Society*, Series C, 54, 507–554. [165]

Rue, H., Martino, S., and Chopin, N. (2009), "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations," *Journal of the Royal Statistical Society*, Series B, 71, 319–392. [164]

Wood, S. N., and Fasiolo, M. (2017), "A Generalized Fellner-Schall Method for Smoothing Parameter Estimation With Application to Tweedie Location, Scale and Shape Models," *Biometrics,* to appear. *arXiv preprint arXiv:1606.04802*. [164,165]

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2017, VOL. 112, NO. 517, Theory and Methods
http://dx.doi.org/10.1080/01621459.2016.1270051

Rejoinder

M. P. Wand

School of Mathematical and Physical Sciences, University of Technology Sydney, Sydney, Australia, and Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers, Queensland University of Technology (QUT), Brisbane, Australia

1. Mean Field Variational Bayes Iterative Scheme Variants

For a given model and dataset, the overarching goal is to minimize the Kullback–Leibler divergence of the joint posterior density function from the q-density function subject to a particular product restriction. This is mean field variational Bayes (MFVB). Apart from the traditional iterative scheme, listed as Algorithm 1 of Ormerod and Wand (2010), there were at least three other alternative MFVB iterative schemes in existence before this discussion took place—given in Winn and Bishop (2005), Minka (2005) and Minka and Winn (2008) under the label of variational message passing (VMP). It could be argued that there is no need for another term (and another acronym) and all approaches are just iterative scheme variants of MFVB. After all, the related approximation approach known as expectation propagation only has one name regardless of the iterative scheme used to obtain its version of optimal q-densities.

The comment of Reiss and Goldsmith offers two more MFVB iterative scheme variants: one without stochastic node to factor messages (Loop B) and one with factor to stochastic node messages replaced by factor-specific subcomponents at the end of their sec. 2. These variants provide interesting enlightenment regarding VMP-type approaches to MFVB. Nevertheless, their comment makes no mention of the arbitrarily large model viewpoint and compartmentalization of MFVB iterations via factor graph fragments, which is the main point of Wand (2017). Their claim that one could "jettison the message passing metaphor altogether" is made in the context of MFVB/VMP for a fixed factor graph. But the more relevant issue is whether this MFVB iterative scheme variant allows for compartmentalization of the algebra and computing in the same way that the Minka (2005) approach does. Put another way, can sec. 4.1 and 5.1– 5.3 of Wand (2017) be reexpressed in terms of factor-specific subcomponents with compartmentalization preserved?

2. Exponential Family Density Functions and Conjugacy

There is no doubt that exponential family density functions and conjugacy play an important role in practical MFVB/VMP. Discussion and additional insight on this aspect appears in the comments of Reiss and Goldsmith, Tran and Blei, and Tu. Each of the fragments presented in sec. 4 of Wand (2017) involve exponential family density functions and conjugate sub-graphs.

Nevertheless, exponential density functions and conjugacy are not intrinsic to MFVB/VMP. Equations (7)– (9) of Wand (2017) apply to messages of *any* form and can be used to compartmentalize MFVB iterative schemes for more elaborate model components such as Negative Binomial and t-distribution likelihoods and non-Gaussian penalization of random effects. Current work with Matt McLean is concerned with fragments for elaborate distribution likelihoods. It will allow for the replacement of updates presented in sec. 4.1.5 of Wand (2017) for Gaussian likelihood semiparametric regression models with similar models having other likelihoods. As with the fragments in Wand (2017), the forthcoming McLean and Wand fragment updates only need to be implemented once within a VMP software project.

3. Distributed Computing

The comments of Reiss and Goldsmith and Tran and Blei include discussion on distributed computing. I was fascinated to read about Pearl (1982), which demonstrates that ideas such as

CONTACT M. P. Wand *matt.wand@uts.edu.au* School of Mathematical and Physical Sciences, University of Technology Sydney, P. O. Box 123, Broadway 2007, Australia.

asynchronous updating for inference in large Bayesian networks go back at least 35 years.

As mentioned in sec. 1 of Wand (2017), Luts (2015) developed an approach to semiparametric regression on distributed datasets using MFVB approximate inference. Expectation propagation approaches to semiparametric regression and related models have also received significant attention recently, with distributed computing being one of the driving forces. Interesting yet-to-be-published work on this front includes Gelman et al. (2014), described in the Tran and Blei comment, and Kim and Wand (2017).

4. Accuracy/Speed Trade-Offs

Wand (2017) is on the interface between statistics and computer science and inevitably gets caught up in differing philosophies of these areas. One point of contention involves trade-offs between accuracy and speed. Biostatistician Tu says "it seems reasonable to demand a more detailed understanding of VMP's capacity for valid inference" and "inference is what science demands. A proven ability to produce valid inference is always expected of a new method," while computer scientists Tran and Blei say "we need algorithms along the frontier, where a user can explicitly define a computational budget and employ an algorithm achieving the best statistical properties within that budget" in accordance with the recent statistics journal article, Jordan (2013), that advocates the incorporation of runtime into statistical risk measures. My work on fast approximate inference is driven by the belief that as the sizes of datasets and models continue to grow, alternatives to accurate but computationally intensive approaches are useful. Examples include applications for which interpretation matters more than valid inference, applications where speed is paramount, the problem of sifting through a large set of candidate models and experimental design. For the last of these, Ryan et al. (2016) is a recent review article with some mention of MFVB.

More than 30 years ago statisticians Breiman et al. (1984) developed classification and regression trees. It has since become a mainstay in business analytics and many other applications due to its speed, ability to handle messy data and interpretability—despite it lacking a proven ability to produce valid inference. Do all of the semiparametric regression methods that Tu uses in his impressive biomedical research have a proven ability to produce valid inference? In 30 years from now, will biomedical researchers be able to afford such a restriction to make the best use of the very large amounts of data at their disposal? Despite its imperfections, variational inference is a principled and versatile paradigm that can be of great use for confronting upcoming deluges of data.

There is scope for improvements in the accuracy of VMP. Section 5.1 of Knowles and Minka (2011) describes two alternatives to the Jaakkola-Jordan device for the logistic likelihood fragment that can lead to more accurate inference, as demonstrated by their Figure 1. One of them is simply the Knowles–Minka–Wand updates given in sec. 5.3 of Wand (2017) but with the Poisson likelihood replaced by its logistic counterpart. Current work with Tui Nolan is concerned with addressing numerical issues that arise with these more accurate logistic likelihood fragments and describing them within the Wand (2017) framework.

5. Model Sparsity

I agree with Wood that model sparsity and sparse matrix methods are very important for semiparametric regression as datasets and models continue to grow in size. As mentioned in Wand (2017), sparse matrix-type refinement (referred to there as matrix algebraic streamlining) of the fragment updates relevant to grouped data is still required for efficient handling of large longitudinal and multilevel datasets. This is a tedious endeavor and lacking in statistical glamor, which can make publication in good statistics journals more challenging. I am grateful to Simon Wood and co-authors for their continual recognition of the importance of numerical analysis in practical semiparametric regression and the solutions provided in their several articles and software such as the mgcv package (Wood 2016) in Wood and Fasiolo (2016) is the latest in an impressive sequence of contributions of this type.

6. Software

Tu, Tran and Blei and Wood each mention software. As discussed in Wand (2017) there is already one software product, Infer.NET, that uses VMP for fast approximate inference. A newer one is BayesPy (Luttinen 2016) for the Python computing environment. The niche that Wand (2017) endeavors to carve out stems from the fact that general purpose software products are limited in terms of how well they handle specific classes of models. Wand (2017) explained the ideas of VMP in statistical contexts and introduces factor graph fragments for arbitrarily large models. It puts this still very young methodology into the hands of statistical programmers who want to develop their own suite of programs for carrying out fast approximate inference for models relevant to their particular data analytic problems. In the case of my main area of research, semiparametric regression, there is enormous potential for software packages that allow large models to be fit quickly to massive datasets. In Lee and Wand (2016a, 2016b), we show how MFVB for large longitudinal and multilevel semiparametric regression analysis can be optimized for speed by streamlining the matrix algebra, and this work can be transferred to the VMP framework. As mentioned by Tran and Blei, parallelization of the required computations is also a possibility.

Acknowledgment

The author thanks each of the discussants for their thoughtful and thought-provoking comments.

ORCID

M. P. Wand http://orcid.org/0000-0003-2555-896X

Additional References

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*, Belmont, CA: Wadsworth Publishing. [167]

Kim, A. S. I., and Wand, M. P. (2017). "On Expectation Propagation for Generalized, Linear and Mixed Models," *Australian and New Zealand Journal of Statistics*, in press. [167]

Luttinen, J. (2016), "BayesPy: Variational Bayesian Inference in Python," *Journal of Machine Learning Research*, 17, 1–6. [167]

Ryan, E. G., Drovandi, C. C., McGree, J. M., and Pettitt, A. N. (2016), "A Review of Modern Computational Algorithms for Bayesian Optimal Design," *International Statistical Review*, 84, 128–154. [167]

Wood, S. N. (2016), "mgcv: Mixed GAM Computation Vehicle With GCV/AIC/REML Smoothness Estimation," R package version 1.8. Available at *http://cran.r-project.org* [167]