Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data

Andrew Gelman* Aki Vehtari† Pasi Jylänki‡ Tuomas Sivula† Dustin Tran*
Swupnil Sahai* Paul Blomstedt† John P. Cunningham* David Schiminovich§
Christian Robert¶

7 Mar 2017

Abstract

A common approach for Bayesian computation with big data is to partition the data into smaller pieces, perform local inference for each piece separately, and finally combine the results to obtain an approximation to the global posterior. Looking at this from the bottom up, one can perform separate analyses on individual sources of data and then want to combine these in a larger Bayesian model. In either case, the idea of distributed modeling and inference has both conceptual and computational appeal, but from the Bayesian perspective there is no general way of handling the prior distribution: if the prior is included in each separate inference, it will be multiply-counted when the inferences are combined; but if the prior is itself divided into pieces, it may not provide enough regularization for each separate computation, thus eliminating one of the key advantages of Bayesian methods. To resolve this dilemma, expectation propagation (EP) has been proposed as a prototype for distributed Bayesian inference. The central idea is to factor the likelihood according to the data partitions, and to iteratively combine each factor with an approximate model of the prior and all other parts of the data, thus producing an overall approximation to the global posterior at convergence.

In this paper, we give an introduction to EP and an overview of some recent developments of the method, with particular emphasis on its use in combining inferences from partitioned data. In addition to distributed modeling of large datasets, our unified treatment also includes hierarchical modeling of data with a naturally partitioned structure.

1. Introduction

Expectation propagation (EP) is a fast and parallelizable method of distributional approximation via data partitioning. In its classical formulation, EP is an iterative approach to approximately minimizing the Kullback-Leibler divergence from a target density $f(\theta)$, to a density $g(\theta)$ from a tractable family. Since its introduction by Opper and Winther (2000) and Minka (2001b), EP has become a mainstay in the toolbox of Bayesian computational methods for inferring intractable posterior densities.

Motivated by the substantial methodological progress made in the last decade and a half, our aim in this paper is to review the current state of the art, also serving readers with no previous exposure to EP as an introduction to the methodology. In our review, we focus on two sets of developments in the literature, which we believe are of particular importance: (i) algorithmic improvements for making the method faster and numerically stable in working with real problems, and (ii) the use of EP as a prototype for message passing in distributed Bayesian inference.

The latter point, which is the main theme of our paper, is treated in the general setting of combining inferences on data partitioned into disjoint subsets. This setting can be motivated from

*Department of Statistics, Columbia University, New York.
†Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Finland.
‡Donders Institute for Brain, Cognition, and Behavior, Radboud University Nijmegen, Netherlands.
§Department of Astronomy, Columbia University, New York.
¶Université Paris Dauphine.

two complementary views, "top-down" and "bottom-up," both of which have gained increasing attention in the statistics and machine learning communities. We approach them as instances of the same computational framework.

The *top-down* view deals with fitting statistical models to large datasets, for which many distributed (divide-and-conquer) algorithms have been proposed over the past few years (Ahn et al., 2012; Balan et al., 2014; Hoffman et al., 2013; Scott et al., 2016; Wang and Dunson, 2013; Neiswanger et al., 2014). The basic idea is to partition the data y into K pieces, $y_1, \ldots, y_K$, each with likelihood $p(y_k|\theta)$, then analyze each part of the likelihood separately, and finally combine the K pieces to perform inference (typically approximately) for θ.

In a Bayesian context, though, it is not clear how distributed computations can handle the prior distribution. If the prior $p(\theta)$ is included in each separate inference, it will be multiply-counted when the K inferences are combined. To correct for this, one can in principle divide the combined posterior by $p(\theta)^{K-1}$ at the end, but this can lead to computational instabilities. An alternative is to divide the prior itself into pieces, but then the fractional prior $p(\theta)^{1/K}$ used for each separate computation may be too weak to effectively regularize, thus eliminating one of the key computational advantages of Bayesian inference, for examples in which the likelihood alone does not allow good estimation of θ; see Gelman et al. (1996), Gelman et al. (2008), and, in the likelihood-free context, Barthelmé and Chopin (2014).

Turning to the *bottom-up* view, we may be motivated to combine information across local sources of data and models. Here the data—not necessarily big in size—are already split into K pieces, each with likelihood $p(y_k|\theta)$. For example, in privacy-preserving computing, the data owners of local pieces can only release aggregated information such as moments (e.g. Sarwate et al., 2014; Dwork and Roth, 2014). In meta-analysis, the different pieces of information come from different sources or are reported in different ways, and the task is to combine such information (Dominici et al., 1999; Higgins and Whitehead, 1996). In both settings, we would like to partially pool information across separate analyses, enabling more informed decisions both globally and to the local analyses. These types of problems fall into the general framework of hierarchical models, and—as in the privacy-preserving setting—may need to be solved without complete access to the local data or model.

Extracting the core principles behind EP motivates a general framework for passing information between inferences on partitioned data (Xu et al., 2014; Hasenclever et al., 2015). We use the idea of a *cavity distribution*, which approximates the effect of inferences from all other $K-1$ data partitions, as a prior in the inference step for individual partitions. In classical EP, the data are usually partitioned pointwise, with the approximating density fully factorized and with additional algorithmic considerations fixed. By partitioning the data into bigger subsets, the same idea can be used in a more versatile manner. While Xu et al. (2014) and Hasenclever et al. (2015) focus on a particular EP algorithm with distributed computation in mind, in this paper, we emphasize the generality of the message passing framework, conforming to both the top-down and bottom-up views. In particular, we present an efficient distributed approach for hierarchical models, which by construction partition the data into conditionally separate pieces. By applying EP to the posterior distribution of the shared parameters, the algorithm's convergence only needs to happen on this parameter subset. We implement an example algorithm using the Stan probabilistic programming language (Stan Development Team, 2016); we leverage its sample-based inferences for the individual partitions.

The remainder of the paper proceeds as follows. We first review the basic EP algorithm and introduce terminology in Section 2. In Section 3, we discuss the use of EP as a general message passing framework for partitioned data, and in Section 4, we further demonstrate its applicability for hierarchical models. Despite being conceptually straightforward, the implementation of an EP

algorithm involves consideration of various options in carrying out the algorithm. In Section 5, we discuss such algorithmic considerations at length, also highlighting recent methodological developments and suggesting further generalizations. Section 6 demonstrates the framework with two hierarchical experiments, and Section 7 concludes the paper with a discussion. Further details of implementation can be found in Appendix A.

2. Expectation propagation

2.1. Basic algorithm

Expectation propagation (EP) is an iterative algorithm in which a target density $f(\theta)$ is approximated by a density from some specified parametric family $g(\theta)$. First introduced by Opper and Winther (2000) and, shortly after, generalized by Minka (2001b,a), EP belongs to a group of *message passing algorithms*, which infers the target density using a collection of localized inferences (Pearl, 1986). In the following, we introduce the general message passing framework and then specify the features of EP.

Let us first assume that the target density $f(\theta)$ has some convenient factorization up to proportion,

$$f(\theta) \propto \prod_{k=0}^{K} f_k(\theta).$$

In Bayesian inference, the target f is typically the posterior density $p(\theta|y)$, where one can assign for example one factor as the prior and other factors as the likelihood for one data point. A message passing algorithm works by iteratively approximating $f(\theta)$ with a density $g(\theta)$ which admits the same factorization,

$$g(\theta) \propto \prod_{k=0}^{K} g_k(\theta),$$

and using some suitable initialization for all $g_k(\theta)$. The factors $f_k(\theta)$ together with the associated approximations $g_k(\theta)$ are referred to as *sites*.

At each iteration of the algorithm, and for $k = 1, \ldots, K$, we take the current approximating function $g(\theta)$ and replace $g_k(\theta)$ by the corresponding factor $f_k(\theta)$ from the target distribution. Accordingly, (with slight abuse of the term "distribution") we define the *cavity distribution*,

$$g_{-k}(\theta) \propto \frac{g(\theta)}{g_k(\theta)},$$

and the *tilted distribution*,

$$g_{\backslash k}(\theta) \propto f_k(\theta) g_{-k}(\theta).$$

The algorithm proceeds by first constructing an approximation $g^{\text{new}}(\theta)$ to the tilted distribution $g_{\backslash k}(\theta)$. After this, an updated approximation to the target density's $f_k(\theta)$ can be obtained as $g_k^{\text{new}}(\theta) \propto g^{\text{new}}(\theta)/g_{-k}(\theta)$. Iterating these updates in sequence or in parallel gives the following algorithm.

General message passing algorithm

1. Choose initial site approximations $g_k(\theta)$.
2. Repeat for $k \in \{1, 2, \ldots, K\}$ (in serial or parallel batches) until all site approximations $g_k(\theta)$ converge:
   (a) Compute the cavity distribution, $g_{-k}(\theta) \propto g(\theta)/g_k(\theta)$.
   (b) Update site approximation $g_k(\theta)$ so that $g_k(\theta)g_{-k}(\theta)$ approximates $f_k(\theta)g_{-k}(\theta)$.

In some sources, step 2b above is more strictly formulated as

$$g_k^{\text{new}}(\theta) = \arg\min \mathrm{D}(f_k(\theta)g_{-k}(\theta) \| g_k(\theta)g_{-k}(\theta)),$$

where $\mathrm{D}(\cdot\|\cdot)$ corresponds to any divergence measure. In our definition, the algorithm can more freely implement any approximation method, which does not necessarily minimize any divergence. For example, classical message passing performs step 2b exactly to get the true tilted distribution (Jordan, 2003).

2.2. Further considerations

In EP, the target pieces $f_k(\theta)$ and site approximations $g_k(\theta)$ are restricted to be in an exponential family such as multivariate normal. This makes the algorithm efficient: any product and division between such distributions stays in the parametric family and can be carried out analytically by summing and subtracting the respective natural parameters. The complexity of these distributions, which is determined by the number of parameters in the model, remains constant regardless of the number of sites. This is less expensive than carrying around the full likelihood, which in general requires computation time proportional to the size of the data. Accordingly, EP tends to be applied to specific high-dimensional problems where computational cost is an issue, notably for Gaussian processes (Rasmussen and Williams, 2006; Jylänki et al., 2011; Cunningham et al., 2011; Vanhatalo et al., 2013), and efforts are made to keep the algorithm both stable and fast.

Approximating the tilted distribution in step 2b is, in many ways, the core step of a message passing algorithm. In EP, this is done by matching the moments of $g_k(\theta)g_{-k}(\theta)$ to those of $f_k(\theta)g_{-k}(\theta)$, which corresponds to minimizing the Kullback-Leibler divergence $\mathrm{KL}(g_{\backslash k}(\theta)||g(\theta))$. In Section 5.1, we discuss in more detail a variety of other choices for forming tilted approximations, also beyond the standard choices in the EP literature.

Even though EP minimizes local KL-divergences in the scope of each site, it is not guaranteed that the KL-divergence from the target density to the global approximation, $\mathrm{KL}(f(\theta)||g(\theta))$, will be minimized. Furthermore, there is no guarantee of convergence for EP in general. However, for models with log-concave factors f_k and initialization to the prior distribution, the algorithm has proven successful in many applications.

Generally, message passing algorithms require that the site distributions $g_k(\theta)$ are stored in memory, which may be a problem with a large number of sites. Dehaene and Barthelme (2015) and Li et al. (2015) present a modified EP method in which each site shares the same site distribution $g_{\text{site}}(\theta)$. While making the algorithm more memory efficient, they show for certain applications that the method works almost as well as the original EP.

Instead of the local site updating scheme of the message passing algorithm, various methods have been developed for directly optimizing the global objective function for EP. Heskes and Zoeter

(2002) and Opper and Winther (2005) present a double loop min-max-based optimization algorithm with guaranteed convergence but possibly far slower than message passing algorithms. Hasenclever et al. (2015) further leverage natural stochastic gradients based on properties of the exponential family in order to speed up this optimization, while also adopting the possibility for α-divergence minimization. Hernández-Lobato et al. (2016) provide a scalable black-box optimization algorithm using stochastic gradients and automatic differentiation.

3. Message passing framework for partitioned data

The factorized nature of the EP algorithm defined in Section 2 makes it a suitable tool for partitioned data. Assuming the likelihood factorizes over the partitions, the likelihood of each part can be assigned for one site. The algorithm can be run in a distributed setting consisting of a central node and site nodes. The central node stores the current global approximation and controls the messaging for the sites, while each site node stores the corresponding data and the current site approximation. The central node initiates updates by sending the current global approximation to the sites. Given this information, a site node can update the site approximation and send back the difference. The central node then receives the differences and aggregates to update the global approximation. This enables model parallelism—in that each site node can work independently to infer its assigned part of the model—and data parallelism—in that each site node only needs to store its assigned data partition (Dean et al., 2012).

In a conventional EP setting, the likelihood is factorized pointwise so that each site corresponds to one data point. This is motivated by the simplicity of the resulting site updates, which can often be carried out analytically. By assigning multiple data points to one site, the updates become more difficult and time consuming. However, updating such a site also provides more information to the global approximation and the algorithm may converge in fewer iterations. In addition, the resulting approximation error is smaller as the number of sites decreases.

As mentioned in Section 2, in EP approximating the tilted distribution in step 2b of the general message passing algorithm is carried out by moment matching. This makes EP particularly useful in the context of partitioned data: intractable site updates can be conveniently inferred by estimating the tilted distribution moments for example with MCMC methods. Nevertheless, other message passing algorithms, where some other method for tilted distribution approximation is used, can also be applied in such a context. These are discussed in more detail in Section 5.1.

In divide-and-conquer algorithms, each partition of the data is processed separately and the results are combined together in a single pass. This behavior resembles the first iteration of the EP algorithm. In EP however, the global approximation is further optimized by iteratively updating the sites with shared information from the other sites. In contrast to divide-and-conquer algorithms, each step of an EP algorithm combines the likelihood of one partition with the cavity distribution representing the rest of the available information across the other $K-1$ pieces (and the prior). This extra information can be used to concentrate the computational power economically in the areas of interest. Figure 1 illustrates this advantage with a simple example. Furthermore, Figure 2 illustrates the construction of the tilted distribution $g_{\backslash k}(\theta)$ and demonstrates the critically important regularization attained by using the cavity distribution $g_{-k}(\theta)$ as a prior; because the cavity distribution carries information about the posterior inference from all other $K-1$ data pieces, any computation done to approximate the tilted distribution (step 2b in the message passing algorithm) will focus on areas of greater posterior mass.

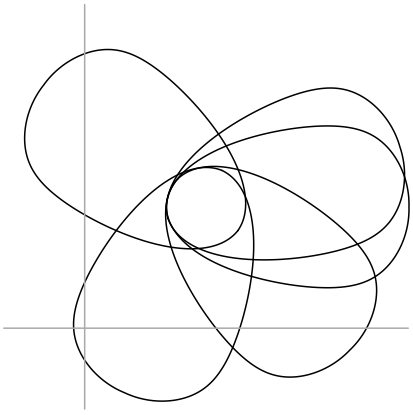

Figure 1: *Sketch illustrating the benefits of message passing in Bayesian computation. In this simple example, the parameter space θ has two dimensions, and the data have been split into five pieces. Each oval represents a contour of the likelihood $p(y_k|\theta)$ provided by a single partition of the data. A simple parallel computation of each piece separately would be inefficient because it would require the inference for each partition to cover its entire oval. By combining with the cavity distribution $g_{-k}(\theta)$, we can devote most of our computational effort to the area of overlap.*

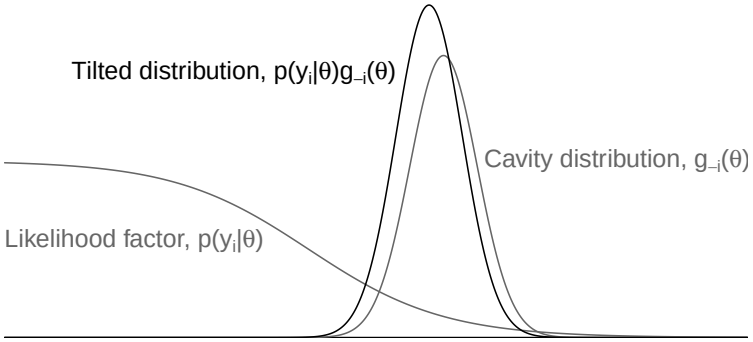

Figure 2: *Example of a step of an EP algorithm in a simple one-dimensional example, illustrating the stability of the computation even when part of the likelihood is far from Gaussian. When performing inference on the likelihood factor $p(y_k|\theta)$, the algorithm uses the cavity distribution $g_{-k}(\theta)$ as a prior.*

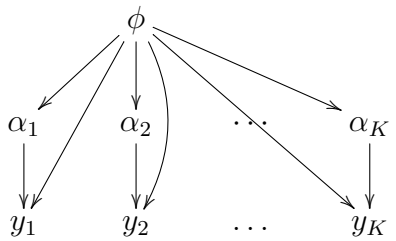

Figure 3: *Model structure for the hierarchical EP algorithm. In each site k, inference is based on the local model, $p(y_k|\alpha_k,\phi)p(\alpha_k|\phi)$, multiplied by the cavity distribution $g_{-k}(\phi)$. Computation on this tilted posterior gives a distributional approximation on (α_k,ϕ) or simulation draws of (α_k,ϕ); in either case, we just use the inference for ϕ to update the local approximation $g_k(\phi)$. The algorithm has potentially large efficiency gains because, in each of the K sites, both the sample size and the number of parameters scale proportional to $1/K$.*

4. Application to hierarchical models

In a hierarchical context, EP can be used to efficiently divide a multiparameter problem into sub-problems with fewer parameters. If the data assigned to one site are not affected by some parameter, the site does not need to take this parameter into account in the update process. By distributing hierarchical groups into separate sites, the sites can ignore the local parameters from the other groups.

4.1. Posterior inference for the shared parameters

Suppose a hierarchical model has local parameters $\alpha_1,\alpha_2,\ldots,\alpha_K$ and shared parameters ϕ. All these can be vectors, with each α_k applying to the model for the data piece y_k, and with ϕ including shared parameters ("fixed effects") of the data model and hyperparameters as well. This structure is displayed in Figure 3. Each data piece y_k is assigned to one site with its own local model $p(y_k|\alpha_k,\phi)p(\alpha_k|\phi)$. As each local parameter α_k affects only one site, they do not need to be included in the propagated messages. EP can thus be applied to approximate the marginal posterior distribution of ϕ only. If desired, the joint posterior distribution of all the parameters can be approximated from the obtained marginal approximation with the methods discussed later in Section 4.2.

Applying EP for the marginal posterior distribution of ϕ is straightforward. The target density factorizes as

$$\int_\alpha p(\phi,\alpha|y)\,\mathrm{d}\alpha \propto p(\phi)\prod_{k=1}^{K}\int_{\alpha_k} p(y_k|\alpha_k,\phi)p(\alpha_k|\phi)\,\mathrm{d}\alpha_k,$$

where $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_K)$. Given the cavity distribution $g_{-k}(\phi)$, each site k approximates the tilted distribution,

$$g_{\backslash k}(\phi) \propto \int_{\alpha_k} g_{-k}(\phi)p(y_k|\alpha_k,\phi)p(\alpha_k|\phi)\,\mathrm{d}\alpha_k,$$

in the restricted exponential family form by determining its moments, after which the site updates the respective approximation $g_k(\phi)$ accordingly. For intractable tilted distributions, as is often the case, simulation based methods provide a practical general approach.

The computational advantage of this marginalized approach is that the local parameters α are partitioned. For example, suppose we have a model with 100 data points in each of 3 000 groups, 2 local parameters per group (a varying slope and intercept) and, say, 20 shared parameters

(including fixed effects and hyperparameters). If we then divide the problem into $n = 3\,000$ pieces, we have reduced a $300\,000 \times 6\,020$ problem to $3\,000$ parallel 100×22 problems. To the extent that computation costs are proportional to sample size multiplied by number of parameters, this is a big win.

4.2. Posterior inference for the joint distribution

Once convergence has been reached for the approximate distribution $g(\phi)$, we approximate the joint posterior distribution by

$$g(\alpha_1, \ldots, \alpha_K, \phi) = g(\phi) \prod_{k=1}^{K} p(y_k|\alpha_k, \phi)p(\alpha_k|\phi).$$

We can work with this expression in one of two ways, making use of the ability to perform inference on the tilted distributions, for example by sampling using Stan.

First, if all that is required are the separate (marginal) posterior distributions for each α_k vector, we can take these directly from the corresponding tilted distribution inferences from the last iteration, which can be assumed to be at approximate convergence. Depending on the method, this will give us simulations of α_k or an approximation to $p(\alpha_k, \phi|y)$.

These separate simulations will not be "in phase," however, in the sense that different sites will reflect different states of ϕ. To get simulation draws from the approximate joint distribution of all the parameters, one can first take some number of draws of the shared parameters ϕ from the EP approximation $g(\phi)$, and then, for each draw, run K parallel processes of Stan to perform inference for each α_k conditional on the sampled value of ϕ. This computation is potentially expensive—for example, to perform it using 100 random draws of ϕ would require 100 separate Stan runs—but, on the plus side, each run should converge fast because it is conditional on the hyperparameters of the model. In addition, it may ultimately be possible to use adiabatic Monte Carlo (Betancourt, 2014) to perform this ensemble of simulations more efficiently.

5. Algorithmic considerations

This section discusses various details related to the implementation of an EP or message passing algorithm in general. Some of the key aspects to consider are:

- **Partitioning the data.** There is always a question of how to partition the data. From the bottom-up view such as with private data, the number of partitions K is simply given by the number of data owners. From the top-down view with distributed computing, K will be driven by computational considerations. If K is too high, the site approximations may not be accurate. But if K is low, then the computational gains will be small. For large problems it could make sense to choose K iteratively, for example starting at a high value and then decreasing it if the approximation seems too poor. Due to the structure of modern computer memory, the computation using small blocks may get additional speed-up if the most of the memory accesses can be made using fast but small cache memory.

- **The parametric form of the approximating distributions $g_k(\theta)$.** The standard choice is the multivariate normal family, which will also work for any constrained space with appropriate transformations; for example, one can use logarithm for all-positive and logit for interval-constrained parameters. For simplicity we may also assume that the prior distribution $p_0(\theta)$ is multivariate normal, as is the case in many practical applications, sometimes after proper

reparameterization. Otherwise, one may treat the prior as an extra site which will also be iteratively approximated by some Gaussian density g_0. In that case, some extra care is required regarding the initialization of g_0. We will discuss alternative options in Section 5.5.

- **Initial site approximations g_k.** One common choice is to use improper uniform distributions. With normal approximation, this corresponds to setting natural parameters to zeros. Alternatively, one could use a broad but proper distribution factored into K equal parts, for example setting each $g_k(\theta) = \mathrm{N}(0, \frac{1}{n}A^2 I)$, where A is some large value (for example, if the elements of θ are roughly scaled to be of order 1, we might set $A = 10$).

- **The algorithm to perform inference on the tilted distribution.** We will discuss three options in Section 5.1: deterministic mode-based approximations, divergence measure minimizations, and Monte Carlo simulations.

- **Asynchronous site updates.** In a distributed context, particularly with unevenly sized data partitions, it can be beneficial to allow a site to be updated as soon as it has finished its previous update, even if some other sites are still busy. Different rules for waiting for more information could be applied here, as long as it is ensured that at least one other site is updated before starting the next iteration.

- **Improper site distributions.** When updating a site term g_k in step 2b in the message passing algorithm, the division by the cavity distribution can yield a covariance or precision matrix that is not positive definite. This is not a problem in itself as the site approximations does not need to be proper distributions. However, improper site distributions may lead into improper global approximations or tilted distributions in the next iterations, which is a problem. Various methods for dealing with this issue are discussed in Section 5.3.

In the following sections, we address a few of these issues in detail, namely, how to approximate the tilted distribution and how to handle potential numerical instabilities in the algorithms.

5.1. Approximating the tilted distribution

In EP, the tilted distribution approximation in step 2b is framed as a moment matching problem, where attention is restricted to approximating families estimable with a finite number of moments. For example, with the multivariate normal family, one chooses the site $g_k(\theta)$ so that the first and second moments of $g_k(\theta)g_{-k}(\theta)$ match those of the possibly intractable tilted distribution $g_{\backslash k}(\theta)$. When applied to Gaussian processes, this approach has the particular advantage that the tilted distribution $g_{\backslash k}(\theta)$ can typically be set up as a univariate distribution over only a single dimension in θ. This dimension reduction implies that the tilted distribution approximation can be performed analytically (e.g. Opper and Winther, 2000; Minka, 2001b) or relatively quickly using one-dimensional quadrature (e.g. Zoeter and Heskes, 2005). In higher dimensions, quadrature gets computationally more expensive or, with a reduced number of evaluation points, the accuracy of the moment computations gets worse. Seeger and Jordan (2004) estimated the tilted moments in multiclass classification using multidimensional quadratures. Without the possibility of dimension reduction in the more general case, approximating the integrals to obtain the required moments over $\theta \in \mathbb{R}^k$ becomes a hard task.

To move towards a black-box message passing algorithm, we inspect the tilted distribution approximation from four perspectives: matching the mode, minimizing a divergence measure, using numerical simulations, and using nested EP. Algorithmically, these correspond to Laplace methods, variational inference, Monte Carlo, and recursive message passing, respectively. Critically, the

resulting algorithms preserve the essential idea that the local pieces of data are analyzed at each step in the context of a full posterior approximation.

Mode-based tilted approximations

The simplest message passing algorithms construct an approximation of the tilted distribution around its mode at each step. As Figure 2 illustrates, the tilted distribution can have a well-identified mode even if the factor of the likelihood does not.

An example of a mode-based approximation is obtained by, at each step, setting g^{new} to be the (multivariate) normal distribution centered at the mode of $g_{\backslash k}(\theta)$, with covariance matrix equal to the inverse of the negative Hessian of $\log g_{\backslash k}$ at the mode. This corresponds to the Laplace approximation, and the message passing algorithm corresponds to Laplace propagation (Smola et al., 2004). The proof presented by Smola et al. (2004) suggests that a fixed point of Laplace propagation corresponds to a local mode of the joint model and hence also one possible Laplace approximation. Therefore, with Laplace approximation, a message passing algorithm based on local approximations corresponds to the global solution. Smola et al. (2004) were able to get useful results with tilted distributions in several hundred dimensions.

The presence of the cavity distribution as a prior (as illustrated in Figure 2) gives two computational advantages to this algorithm. First, we can use the prior mean as a starting point for the algorithm; second, the use of the prior ensures that at least one mode of the tilted distribution will exist.

To improve upon this simple normal approximation, we can evaluate the tilted distribution at a finite number of points around the mode and use this to construct a better approximation to capture asymmetry and long tails in the posterior distribution. Possible approximate families include the multivariate split-normal (Geweke, 1989; Villani and Larsson, 2006), split-t, or wedge-gamma (Gelman et al., 2014) distributions. We are *not* talking about changing the family of approximate distributions g—we would still keep these as multivariate normal. Rather, we would use an adaptively-constructed parametric approximation, possibly further improved by importance sampling (Geweke, 1989; Vehtari et al., 2016) or central composite design integration (Rue et al., 2009) to get a better approximation of the moments of the tilted distribution to use in constructing of g_k.

Variational tilted approximations

Mode-finding message passing algorithms have the advantage of simplicity, but they can do a poor job at capturing uncertainty when approximating the tilted distribution. An alternative approach is to find the closest distribution within an approximating family to the tilted distribution, using a divergence measure to define closeness. If the approximating family contains the tilted distribution as one member in the family, then the local inference is exact (step 2b in the algorithm). In practice, this is not the case, and the behavior of the local variational approximations depends on the properties of the chosen divergence measure. This generalizes mode-finding, which corresponds to minimizing a particular divergence measure.

In the classical setup of EP, the chosen divergence measure is the Kullback-Leibler divergence, $\text{KL}(g_{\backslash k}(\theta)||g^{\text{new}}(\theta))$. As discussed before in Section 2, if the approximating distribution forms an exponential family, minimizing the divergence conveniently corresponds to matching the moments of two distributions (Minka, 2001b).

Another reasonable divergence measure is to consider the reverse KL divergence, $\text{KL}(g^{\text{new}}(\theta)||g_{\backslash k}(\theta))$. This is known as variational message passing (Winn and Bishop, 2005), where the local computations

to approximate the tilted distribution can be shown to maximize a lower bound on the marginal likelihood. In fact, variational message passing enjoys the property that the algorithm minimizes a global divergence to the posterior, $\mathrm{KL}(g(\theta)||p(\theta|y))$, according to the factorized approximating family $g(\theta) = p(\theta)\prod_{k=1}^{K} g_k(\theta)$. This connects to much recent work on variational inference. For example, stochastic variational inference (SVI) (Hoffman et al., 2013) uses data subsampling in order to scale computation of the global parameters, bypassing the fact that global parameter computations depend on inferences for all local parameters (as in Section 4); with attention to models defined by conditionally conjugate exponential families, SVI enables fast stochastic optimization using natural gradients. Black box variational inference (Ranganath et al., 2014) generalizes SVI to the class of probability models with a tractable log joint density, based on taking Monte Carlo estimates for gradient-based optimization. Automatic differentiation variational inference further automates the inference using techniques such as reverse-mode automatic differentiation to calculate gradients, as well as a Gaussian variational family on a transformed unconstrained space (Kucukelbir et al., 2016).

Inference can also be done using the α-divergence family, in which $\alpha = 1$ corresponds to the KL divergence used in the classical EP, $\alpha = 0$ corresponds to the reverse KL divergence, and $\alpha = 0.5$ corresponds to Hellinger distance. One algorithm to solve this is known as power EP (Minka, 2004). Power EP has been shown to improve the robustness of the algorithm when the approximation family is not flexible enough (Minka, 2005) or when the propagation of information is difficult due to vague prior information (Seeger, 2008). This can be useful when moment computations are not accurate, as classical EP may have stability issues (Jylänki et al., 2011). Even with one-dimensional tilted distributions, moment computations are more challenging if the tilted distribution is multimodal or has long tails. Ideas of power EP in general might help to stabilize message passing algorithms that use approximate moments, as α-divergence with $\alpha < 1$ is less sensitive to errors in tails of the approximation.

Simulation-based tilted approximations

An alternative approach is to use simulations (for example, Hamiltonian Monte Carlo using Stan) to approximate the tilted distribution at each step and then use these to set the moments of the approximating family. As above, the advantage of the EP message passing algorithm here is that the computation only uses a fraction $1/K$ of the data, along with a simple multivariate normal prior that comes from the cavity distribution.

Similar to methods such as stochastic variational inference (Hoffman et al., 2013) which take steps based on stochastic estimates, EP update steps require unbiased estimates. When working with the normal approximation, we then need estimates of the mean and covariance or precision matrix of the tilted distribution in step 2b. Section 5.4 discusses the problem of estimating the precision matrix from samples. The variance of the estimates can be reduced while preserving unbiasedness by using control variates. While MCMC computation of the moments may give inaccurate estimates, we suspect that they will work better than, or as a supplement to, Laplace approximation for skewed distributions.

In serial or parallel EP, samples from previous iterations can be reused as starting points for Markov chains or in importance sampling. We discuss briefly the latter. Assume we have obtained at iteration t for node k, a set of posterior simulation draws $\theta^s_{t,k}$, $s = 1, \ldots, S_{t,k}$ from the tilted distribution $g^t_{\backslash k}$, possibly with weights $w^s_{t,k}$; take $w^s_{t,k} \equiv 1$ for an unweighted sample. To progress to node $k+1$, reweight these simulations as: $w^s_{t,k+1} = w^s_{t,k} g^t_{\backslash(k+1)}(\theta^s_{t,k}) / g_{\backslash k}(\theta^s_{t,k})$. If the effective

sample size (ESS) of the new weights,

$$\text{ESS} = \frac{\left(\frac{1}{S}\sum_{s=1}^{S} w_{t,k+1}^{s}\right)^2}{\frac{1}{S}\sum_{s=1}^{S}(w_{t,k+1}^{s})^2},$$

is large enough, keep this sample, $\theta_{t,k+1}^{s} = \theta_{t,k}^{s}$. Otherwise throw it away, simulate new $\theta_{t+1,k}^{s}$'s from $g_{\backslash k+1}^{t}$, and reset the weights $w_{t,k+1}$ to 1. This basic approach was used in the EP-ABC algorithm of Barthelmé and Chopin (2014). Furthermore, instead of throwing away a sample with too low an ESS, one could move these through several MCMC steps, in the spirit of sequential Monte Carlo (Del Moral et al., 2006). Another approach, which can be used in serial or parallel EP, is to use adaptive multiple importance sampling (Cornuet et al., 2012), which would make it possible to recycle the simulations from previous iterations. Even the basic strategy should provide important savings when EP is close to convergence. Then changes in the tilted distribution should become small and as a result the variance of the importance weights should be small as well. In practice, this means that the last EP iterations should essentially come for free.

Nested EP

In a hierarchical setting, the model can be fit using the nested EP approach (Riihimäki et al., 2013), where moments of the tilted distribution are also estimated using EP. This approach leads to recursive message passing algorithms, often applied in the context of graphical models, where the marginal distributions of all the model parameters are inferred by passing messages along the edges of the graph (Minka, 2005) in a distributed manner. As in the hierarchical case discussed in Section 4, the marginal approximation for the parameters can be estimated without forming the potentially high-dimensional joint approximation of all unknowns. This framework can be combined together with other message passing methods, adopting suitable techniques for different parts of the model graph. This distributed and extendable approach makes it possible to apply message passing to arbitrarily large models (Wand, 2017).

5.2. Damping

Although the EP algorithm iteratively minimizes the KL-divergences from the tilted distributions to their corresponding approximations, it does not ensure that the KL-divergence from the the target density to the global approximation is minimized. In particular, running the EP updates in parallel often yields a deviated global approximation when compared to the result obtained with sequential updates (Minka and Lafferty, 2002; Jylänki et al., 2011). In order to fix this problem, damping can be applied to the site approximation updates.

Damping is a simple way of performing an EP update on the site distribution only partially by reducing the step size. Consider a damping factor $\delta \in (0, 1]$. A partially damped update can be carried out by,

$$g_k^{\text{new}}(\theta) = g_k(\theta)^{1-\delta}\left(\widetilde{g}_{\backslash k}(\theta)/g_{-k}(\theta)\right)^{\delta},$$

where $\widetilde{g}_{\backslash k}(\theta)$ is the corresponding tilted distribution approximation. This corresponds to scaling the difference in the natural parameters of $g_k(\theta)$ by δ. When $\delta = 1$, no damping is applied at all.

The error in the parallel EP approximation can be avoided by using a small enough damping factor δ. However, this reduction in the step size makes the convergence slower and thus it is beneficial to keep it as close to one as possible. The amount of damping needed varies from problem to problem and it can often be determined by testing. As a safe rule, one can use $\delta = 1/K$. However,

with a large number of sites K, this often results in intolerably slow convergence. In order to speed up the convergence, it could be possible to start off with smaller damping and increase it gradually with the iterations without affecting the resulting approximation.

In addition to fixing the approximation error, damping helps in dealing with some convergence issues, such as oscillation and non positive definiteness in approximated parameters. If these problems arise with the selected damping level, one can temporarily decrease it until the problem is solved.

5.3. Keeping the covariance matrix positive definite

In EP, it is not required that the site approximations be proper distributions. They are approximating a likelihood factor, not a probability distribution, at each site. Tilted distributions and the global approximation, however, must be proper, and situations where these would become improper must be addressed somehow. These problems can be caused by numerical instabilities and also can also be inherent to the algorithm itself.

As discussed before, obtaining the updated site distribution from an approximated tilted distribution in step 2b of the message passing algorithm, can be conveniently written in terms of the natural parameters of the exponential family:

$$Q_k^{\text{new}} = Q_{\backslash k}^{\text{new}} - Q_{-k}, \qquad r_k^{\text{new}} = r_{\backslash k}^{\text{new}} - r_{-k},$$

where each $Q = \Sigma^{-1}$ denote the precision matrix and each $r = \Sigma^{-1}\mu$ denote the precision mean of the respective distribution. Here the approximated natural parameters $Q_{\backslash k}^{\text{new}}$ and $r_{\backslash k}^{\text{new}}$ of the tilted distribution together with the parameters Q_{-k}^{new} and r_{-k}^{new} of the cavity distribution are being used to determine the new site approximation parameters Q_k^{new} and r_k^{new}. As the difference between the two positive definite matrices is not itself necessarily positive definite, it can be seen that the site approximation can indeed become improper.

Often problems in the tilted distribution occur when many of the site approximations become improper. Constraining the sites to proper distributions (perhaps with the exception of the initial site approximations) often fix some of these problems (Minka, 2001b). In the case of a multivariate normal distribution, this corresponds to forcing the covariance or precision matrix to be positive definite. If all the sites are positive definite, all the cavity distributions and the global approximation should also be positive definite. Although effective in helping with convergence, this method has the downside of ignoring information from the sites.

The simplest way of dealing with non-positive definite matrices is to simply ignore any update that would lead into such and hope that future iterations will fix this issue. Another simple option is to set the covariance matrix $\Sigma_k^{\text{new}} = aI$ with some relatively big a and preserve the mean.

Various methods exist for transforming a matrix to become positive definite. One idea, as in the SoftAbs map of Betancourt (2013), is to do an eigendecomposition, keep the eigenvectors but replace all negative eigenvalues with a small positive number and reconstruct the matrix. Another possibly more efficient method is to find only the smallest eigenvalue of the matrix and add its absolute value and a small constant to all the diagonal elements in the original matrix. The former method is more conservative, as it keeps all the eigenvectors and positive eigenvalues intact, but it is computationally heavy and may introduce numerical error. The latter preserves the eigenvectors but changes all of the eigenvalues. However, it is computationally more efficient. If the matrix is only slightly deviated from a positive definite one, it is justified to use the latter one as the change on the eigenvalues is not big. If the matrix has big negative eigenvalues, it is probably best not to try to modify it in the first place.

If damping is used together with positive definite constrained sites, it is only necessary to constrain the damped site precision matrix, not the undamped one. Because of this, it is possible to find a suitable damping factor δ so that the update keeps the site, or all the sites in parallel EP, positive definite. This can also be used together with other methods, for example by first using damping to ensure that most of the sites remain valid and then modifying the few violating ones. Also one option is to just ignore suspicious site updates and hope that the next iteration succeeds.

5.4. Estimating the natural parameters

When using sample-based methods for the inference on the tilted distribution, one must consider the accuracy of the moment estimation. An efficient EP implementation requires that the tilted distribution parameters are estimated in natural form. However, estimating the precision matrix from a set of samples is a complex task and in general the estimates are biased.

The naive way of estimating the precision matrix Q is to invert the unbiased sample covariance matrix, that is $\widehat{Q} = \widehat{\Sigma}^{-1} = (n-1)S^{-1}$, where S is the scatter matrix constructed from the samples. However, in general this estimator is biased: $\mathrm{E}(\widehat{Q}) \neq Q$. Furthermore, the number of samples n affect the accuracy of the estimate drastically. In an extreme case, when n is less than the number of dimensions d, the sample covariance matrix is not even invertible as its rank can not be greater than n. In such a case, one would have to resort for example to the Moore–Penrose pseudo-inverse. In practice, when dealing with the inverse of the scatter matrix, one should apply the QR-decomposition to the samples in order to obtain the Cholesky decomposition of S without ever forming the scatter matrix itself. This is discussed in more detail in Appendix A.2.

If the tilted distribution is normally distributed, an unbiased estimator for the precision matrix can be constructed by (Muirhead, 2005, p. 136)

$$\widehat{Q}_{\mathrm{N}} = \frac{n-d-2}{n-1}\widehat{\Sigma}^{-1} = (n-d-2)S^{-1}. \tag{1}$$

Furthermore, the precision mean is given by

$$\widehat{r}_{\mathrm{N}} = \widehat{Q}_{\mathrm{N}}\widehat{\mu} = (n-d-2)S^{-1}\widehat{\mu}, \tag{2}$$

which can be solved simultaneously while inverting the scatter matrix. Other improved estimates for the normal distribution and some more general distribution families exist (Bodnar and Gupta, 2011; Gupta et al., 2013; Sarr and Gupta, 2009; Tsukuma and Konno, 2006). Different methods for estimating the precision matrix in the general case, that is when no assumptions can be made about the tilted distribution, have also been proposed. These methods often either shrink the eigenvalues of the sample covariance matrix or impose sparse structure constraints to it (Bodnar et al., 2014; Friedman et al., 2008).

5.5. Different families of approximate distributions

We can place the EP approximation, the tilted distributions, and the target distribution on different rungs of a ladder:

- $g = p_0 \prod_{k=1}^{K} g_k$, the EP approximation;
- For any k, $g_{\setminus k} = g\frac{p_k}{g_k}$, the tilted distribution;
- For any k_1, k_2, $g_{\setminus k_1,k_2} = g\frac{p_{k_1}p_{k_2}}{g_{k_1}g_{k_2}}$, which we might call the tilted^2 distribution;

- For any k_1, k_2, k_3, $g_{\backslash k_1,k_2,k_3} = g\frac{p_{k_1}p_{k_2}p_{k_3}}{g_{k_1}g_{k_2}g_{k_3}}$, the tilted^3 distribution;
- ...
- $p = \prod_{k=0}^{K} p_k$, the target distribution, which is also the tilted^K distribution.

From a variational perspective, expressive approximating families for g, that is, beyond exponential families, could be used to improve the individual site approximations (Tran et al., 2016; Ranganath et al., 2016). Instead of independent groups, tree structures could also be used (Opper and Winther, 2005). Even something as simple as mixing simulation draws from the tilted distribution could be a reasonable improvement on its approximation. One could then go further, for example at convergence computing simulations from some of the tilted distributions.

Message passing algorithms can be combined with other approaches to data partitioning. In the present paper, we have focused on the construction of the approximate densities g_k with the goal of simply multiplying them together to get the final approximation $g = p_0 \prod_{k=1}^{K} g_k$. However, one could instead think of the cavity distributions g_{-k} at the final iteration as separate priors, and then follow the ideas of Wang and Dunson (2013).

Another direction is to compare the global approximation with the tilted distribution, for example by computing a Kullback-Leibler divergence or looking at the distribution of importance weights. Again, we can compute all the densities analytically, we have simulations from the tilted distributions, and we can trivially draw simulations from the global approximation, so all these considerations are possible.

6. Experiments

6.1. Simulated hierarchical logistic regression

We demonstrate the distributed EP algorithm with a simulated hierarchical logistic regression problem:

$$y_{ij} \,|\, x_{ij}, \beta_j \sim \text{Bernoulli}\Big(\text{logit}^{-1}(\beta_j^{\mathrm{T}} x_{ij})\Big),$$

where

$$\begin{aligned}
\beta_{jd} &\sim \mathrm{N}(\mu_d, \sigma_d^2), \\
\mu_d &\sim \mathrm{N}(0, \tau_\mu^2), \\
\sigma_d &\sim \text{log-N}(0, \tau_\sigma^2),
\end{aligned}$$

for all dimensions $d = 0, 1, \ldots, D$, groups $j = 1, 2, \ldots, J$, and samples $i = 1, 2, \ldots, N_j$. The first element in the vector β_j corresponds to the intercept coefficient and correspondingly the first element in the data vector $x_{i,j}$ is set to one. The shared parameters inferred with EP are $\phi = (\mu, \log \sigma)$. The model is illustrated graphically in Figure 4.

The simulated problem is constructed with a $D = 16$ dimensional explanatory variable resulting in a total of $2(D + 1) = 34$ shared parameters. The number of hierarchical groups is $J = 64$ and the number of samples per group is $N_j = 25$ for all $j = 1, \ldots, J$, resulting in a total of $N = 1600$ samples. The correlated explanatory variable is sampled from a normal distribution $\mathrm{N}(\mu_{x_j}, \Sigma_{x_j})$, where μ_{x_j} and Σ_{x_j} are regulated so that the latent probability $\text{logit}^{-1}(\beta_j^{\mathrm{T}} x_{ij})$ gets values near zero and one with low but not too low frequency. This ensures that the problem is neither too easy nor too hard.

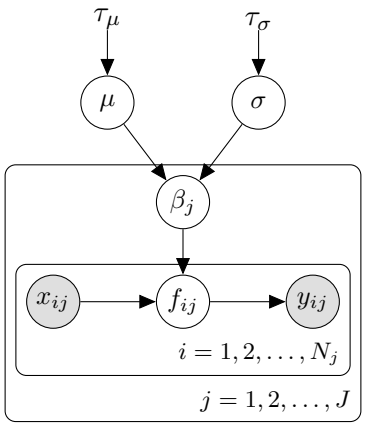

Figure 4: A graphical model representation of the experimented hierarchical logistic regression problem. Indexing $j = 1, 2, \ldots, J$ corresponds to hierarchical groups and $i = 1, 2, \ldots, N_j$ corresponds to observations in group j. Gray nodes represent observed variables and white nodes represent unobserved latent variables. Variables without circles denote fixed priors.

Our implementation uses Python for the message passing framework and the Stan probabilistic modeling language (Stan Development Team, 2016) for MCMC sampling from the tilted distribution. We ran experiments partitioning the data into $K = 2, 4, 8, 16, 32, 64$ sites, using uniform distributions as initial site approximations. For the tilted distribution inference, the natural parameters were estimated with (1) and (2). Each parallel MCMC run had 8 chains with 400 samples, of which half were discarded as warmup. A constant damping factor of $\delta = 0.1$ was used in order to get coherent convergence results amongst different partitions. We compare the results from the distributed EP approximations to an MCMC approximation for the full model using Stan. The full approximation uses 4 chains with 10000 samples, of which half is discarded as warmup. The code for the experiments is available at `https://github.com/gelman/ep-stan`.

If we were to use a simple scheme of data splitting and separate inferences (without using the cavity distribution as an effective prior distribution at each step), the computation would be problematic: with only 25 data points per group, each of the local posterior distributions would be wide, as sketched in Figure 1. The message passing framework, in which at each step the cavity distribution is used as a prior, keeps computations more stable and focused.

Figure 5 illustrates the progression of the experiment for each run. The final approximation quality is better with fewer sites but more sites provide opportunities for faster convergence. This advantage in computation time depends on the implementation of the parallelization. By using the time spent on the sampling of the tilted distribution as our benchmarking criterion, we can focus on the crucial part of the algorithm and neglect the implementational factor.

Figure 6 shows a comparison between posterior mean and standard deviation between the distributed EP approximation and full MCMC approximation for the shared parameters in the extreme cases $K = 2$ and $K = 64$. Points closer to the red diagonal line imply a better EP approximation. It can be seen that the case $K = 2$ results in an overall better approximation. It can also be seen that EP tends to underestimate the variance with more sites. This underestimation is a known feature in EP when there are many sites and strong posterior dependencies (Cunningham et al., 2011).

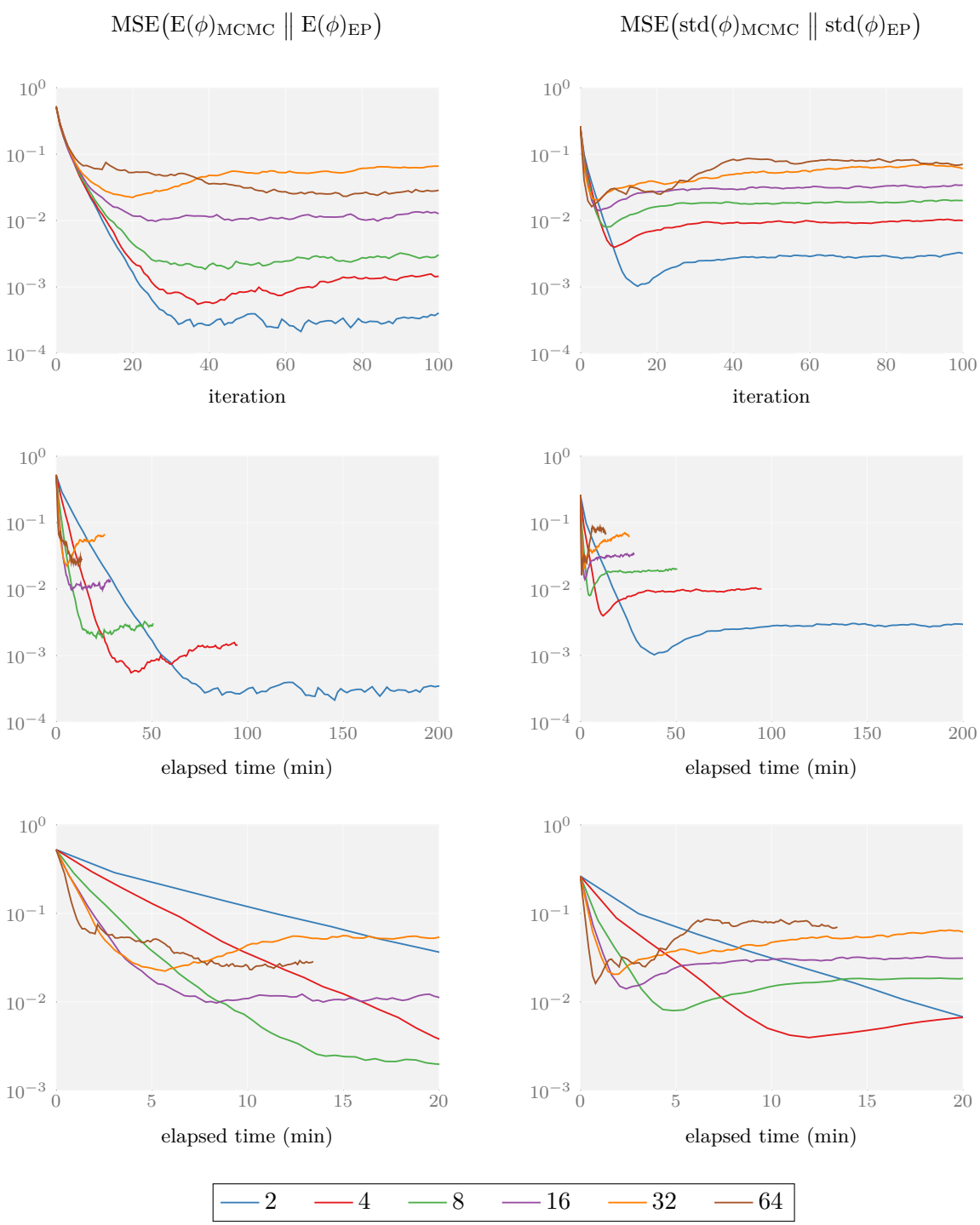

Figure 5: MSE of the posterior mean and standard deviation for full MCMC and the distributed EP approximation for (i) each iteration (top row), and (ii) as a function of the elapsed tilted distribution sampling time (middle and bottom rows). In each graph, the different lines correspond to a partitioning of the data into 2, 4, 8, etc., sites. The y-axis is in the logarithmic scale. Unsurprisingly, the final accuracy declines as the number of partitions increases—but if a bit of approximation is acceptable, the highly parallel algorithms converge rapidly.

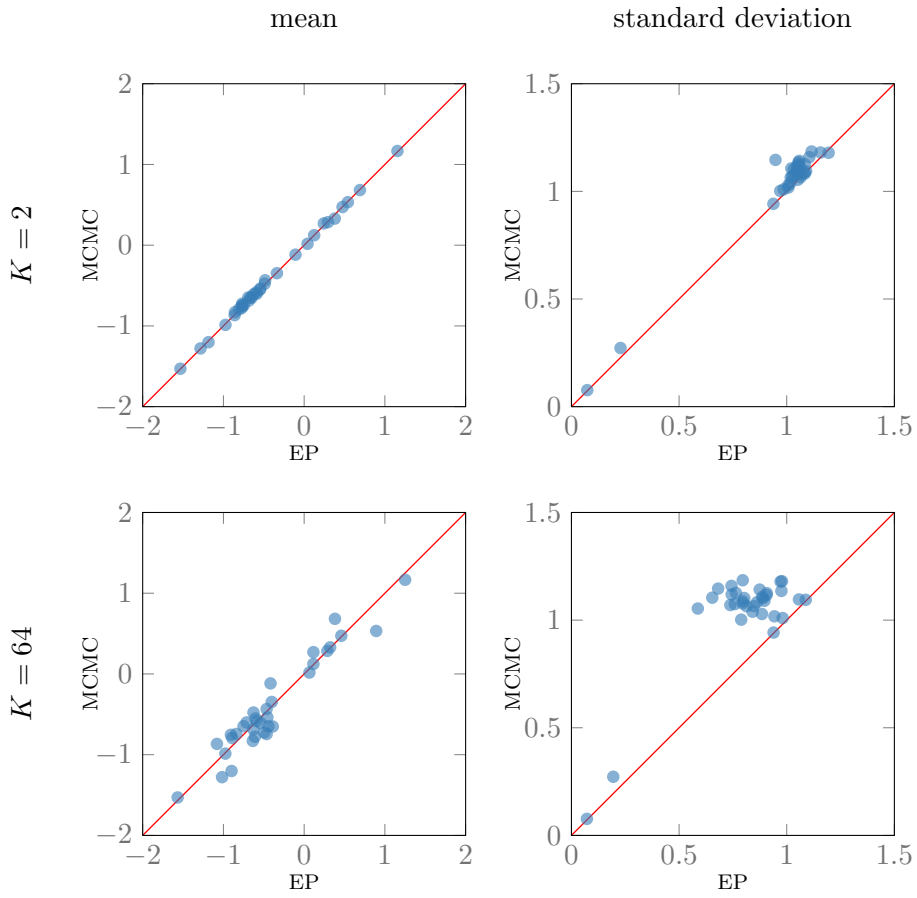

Figure 6: Comparison of the posterior mean and standard deviation of full MCMC and the final distributed EP approximation when the groups are distributed into $K = 2$ (top row) and $K = 64$ (bottom row) sites. Each dot corresponds to one of the 34 shared parameters. The red diagonal line shows the points of equivalence.

6.2. Hierarchical mixture model applied to actual astronomy data

We also demonstrate the distributed EP algorithm applied to an actual data set in astronomy. The goal of our inference is to model the nonlinear relationship between diffuse Galactic far ultraviolet radiation (FUV) and 100-μm infrared emission (i100) in various regions of the observable universe. Data is collected from the Galaxy Evolution Explorer telescope. It has been shown that there is a linear relationship between FUV and i100 below i100 values of 8 MJy sr^{-1} (Hamden et al., 2013). Here we attempt to model this relationship across the entire span of i100 values.

Figure 7 shows scatterplots of FUV versus i100 in different longitudinal regions (each of width 1 degree) of the observable universe. The bifurcation in the scatterplots for i100 values greater than 8 MJy sr^{-1} suggests a non-linear mixture model is necessary to capture the relationship between the two variables. At the same time, a flexible parametric model is desired to handle the various mixture shapes, while maintaining interpretability in the parameters.

Letting $\sigma(\cdot) = \text{logit}^{-1}(\cdot)$ denote the inverse logistic function and letting a_j,

$$a_j = [\beta_{0j}, \beta_{1j}, \mu_{1j}, \sigma_{1j}, \sigma^{-1}(\beta_{2j}), \mu_{2j}, \sigma_{2j}, \sigma^{-1}(\pi_j), \sigma_j]^T,$$

denote the local parameters for each group j, we model the top part of the bifurcation (i.e. the first

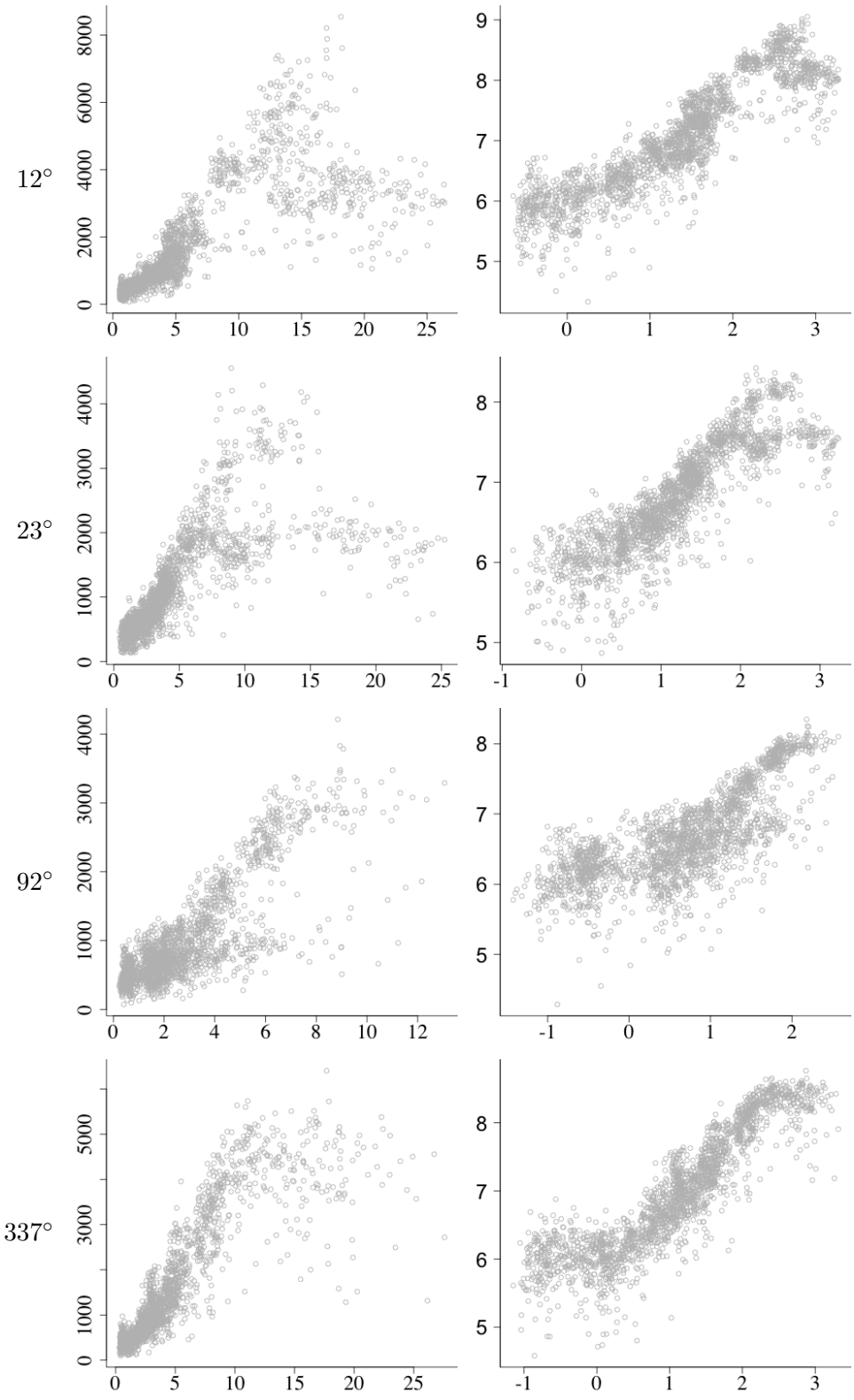

Figure 7: Scatterplots of ultraviolet radiation (FUV) versus infrared radiation (i100) in various regions of the universe. Data are shown for regions of longitude $12°, 23°, 92°$, and $337°$, and are presented with axes on the original scale (first column) and on the log scale (second column).

component of the mixture) as a generalized inverse logistic function,

$$f(a_j, x_{ij}) = \beta_{0j} + \beta_{1j}\sigma\left(\frac{\log x_{ij} - \mu_{1j}}{\sigma_{1j}}\right),$$

while the second mixture component is modeled as the same inverse logistic function multiplied by an inverted Gaussian:

$$g(a_j, x_{ij}) = \beta_{0j} + \beta_{1j}\sigma\left(\frac{\log x_{ij} - \mu_{1j}}{\sigma_{1j}}\right) \cdot \left(1 - \beta_{2j}\exp\left(-\frac{1}{2}\left(\frac{\log x_{ij} - \mu_{2j}}{\sigma_{2j}}\right)^2\right)\right).$$

As such, the ultraviolet radiation (y_{ij}) is modeled as a function of infrared radiation (x_{ij}) through the following mixture model:

$$\begin{aligned} \log y_{ij} &= \pi_j \cdot f(a_j, x_{ij}) + (1 - \pi_j) \cdot g(a_j, x_{ij}) + \sigma_j \epsilon_{ij}, \\ \epsilon_{ij} &\sim N(0, 1), \end{aligned}$$

where $\beta_{2j} \in [0, 1]$, $\pi_j \in [0, 1]$, and the local parameters are modeled hierarchically with the following shared centers and scales:

$$\begin{aligned} \beta_{0j} &\sim \mathrm{N}(\beta_0, \tau^2_{\beta 0}), \\ \beta_{1j} &\sim \mathrm{N}(\beta_1, \tau^2_{\beta 1}), \\ \mu_{1j} &\sim \text{log-N}(\log \mu_1, \tau^2_{\mu 1}), \\ \sigma_{1j} &\sim \text{log-N}(\log \sigma_1, \tau^2_{\sigma 1}), \\ \sigma^{-1}(\beta_{2j}) &\sim \mathrm{N}(\sigma^{-1}(\beta_2), \tau^2_{\beta 2}), \\ \mu_{2j} &\sim \text{log-N}(\log \mu_2, \tau^2_{\mu 2}), \\ \sigma_{2j} &\sim \text{log-N}(\log \sigma_2, \tau^2_{\sigma 2}), \\ \sigma^{-1}(\pi_j) &\sim \mathrm{N}(\sigma^{-1}(\pi), \tau^2_{\pi}), \\ \sigma_j &\sim \text{log-N}(\sigma, \tau^2_{\sigma}) \end{aligned}$$

for all groups $j = 1, 2, \ldots, J$, and samples $i = 1, 2, \ldots, N_j$.

Hence are our problem has $9 \cdot 2 = 18$ shared parameters of interest. The number of local parameters depends on how finely we split the data in the observable universe. Our study in particular is constructed with $J = 360$ hierarchical groups (one for each longitudinal degree of width one degree), resulting in a total of $9 \cdot J = 3,240$ local parameters. We also sample the number of observations per group as $N_j = 2,000$ for all $j = 1, \ldots, J$, resulting in a total of $N = 720,000$ samples.

Our implementation uses R for the message passing framework and the Stan probabilistic modeling language (Stan Development Team, 2016) for MCMC sampling from the tilted distribution. We fit the mixture model with various EP settings, partitioning the data into $K = 5, 10, 30$ sites and using uniform distributions as the initial site approximations. For the tilted distribution inference, the natural parameters are estimated with (1) and (2). Each parallel MCMC run has 4 chains with 1000 iterations, of which half are discarded as warmup. A constant damping factor of $\delta = 0.1$ is used in order to get coherent convergence results amongst different partitions. We compare the results from the distributed EP approximations to an MCMC approximation for the full model using Stan. The full approximation uses 4 chains with 1000 iterations, of which half is discarded as warmup.

Figure 8 illustrates the computation times for the EP runs with both serial and parallel updates. The advantages of distributed EP are most clear when comparing K = 1 site to K = 30 sites, which

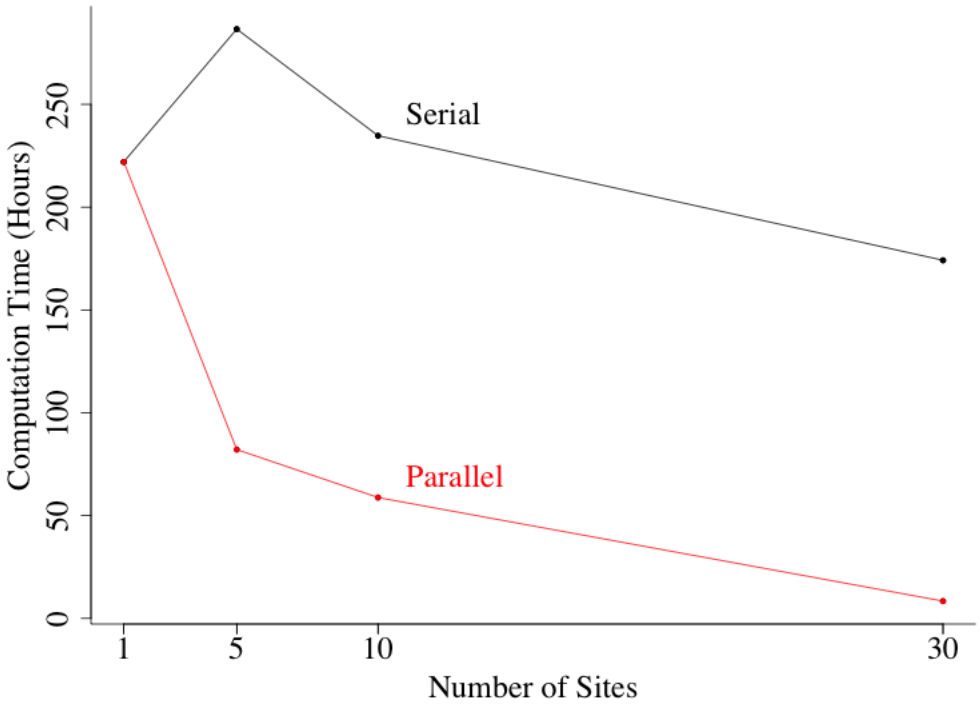

Figure 8: Computation times for the distributed EP algorithm applied to the astronomy data set, as a function of the number of sites. The full MCMC computation time is equivalent to that of EP with $K = 1$ site. The computational benefits of increasing the number of sites is clear when the updates are parallel.

results in a 96% decrease in computation time. This advantage in computation time, however, depends on the implementation of the parallelization. By using the time spent on the sampling of the tilted distribution as our benchmarking criterion, we can focus on the crucial part of the algorithm and neglect the implementation-specific factor.

Figure 9 shows a comparison of the local scatterplot fits for each EP setting on various hierarchical groups, each representing a one-degree longitudinal slice of the observable universe. While all of the runs show similar results for most groups, there are some cases where increasing the number of sites results in poorer performance. In particular, EP with 30 sites converges to a different mixture for 82°, while EP with 10 sites converges to a different mixture for 194°.

7. Discussion

Using the principle of message passing with cavity and tilted distributions, we presented a framework for Bayesian inference on partitioned data sets. Similar to more conventional divide-and-conquer algorithms, EP can be used to divide the computation into manageable sizes without scattering the problem into too small pieces. Further, EP comes with the additional advantage of naturally sharing information between distributed parts, focusing the computation into important areas of the parameter space. From an alternative point of view, EP can also be used to pool information across many sources of already partitioned data sets and models. In the case of hierarchical models, EP enables efficient distributed computation for large models with big data sets, as well as meta-models fit into local models or local aggregated data.

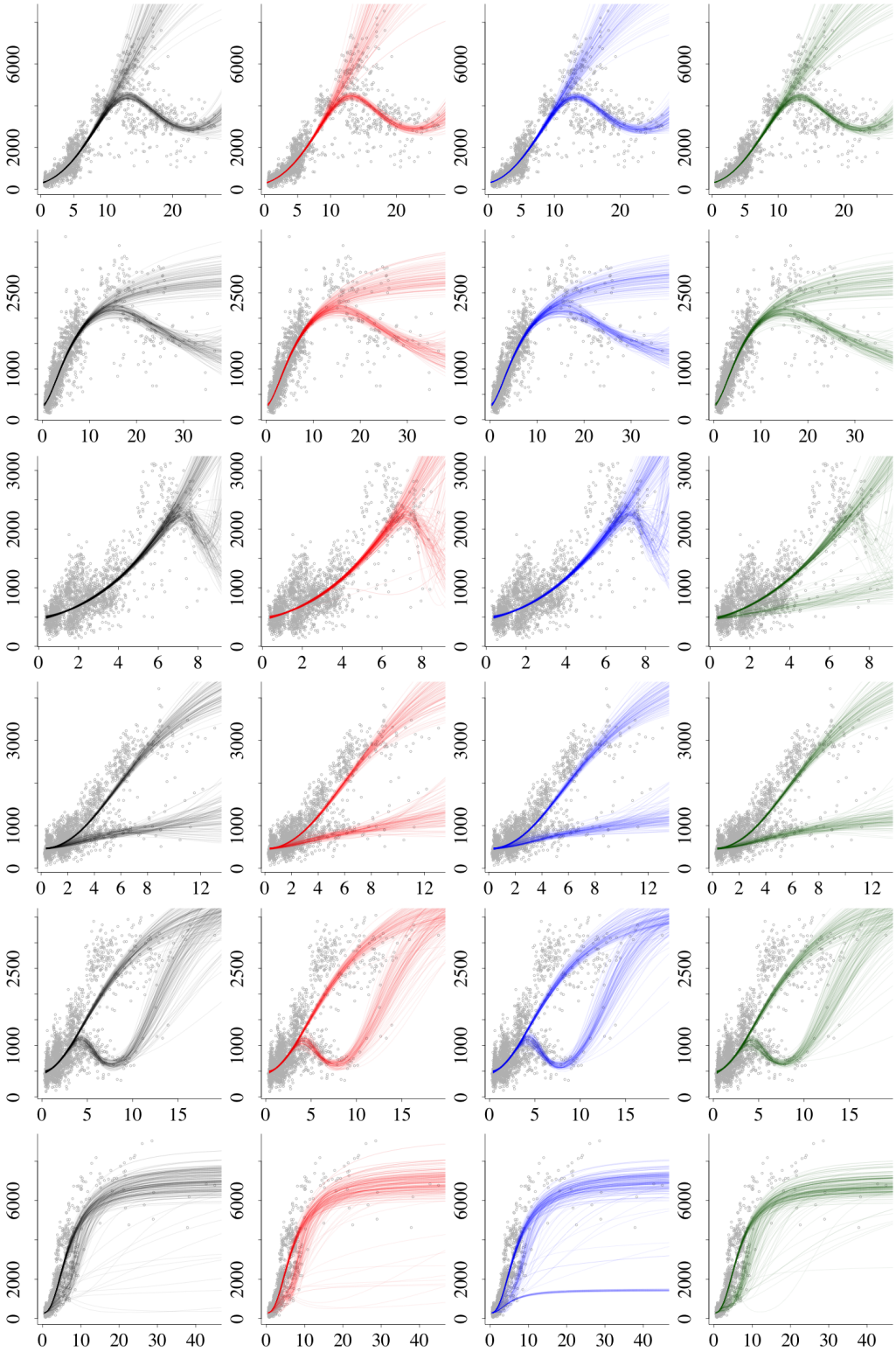

Figure 9: Comparison of the local fits of the full MCMC computation (black) for the astronomy example and the final distributed EP approximations when the groups are distributed into $K = 5$ (red), $K = 10$ (blue), and $K = 30$ (green) sites. Posterior draws are shown for each of 6 groups (one group per row) with longitudes $12°, 32°, 82°, 92°, 93°$, and $194°$.

The message passing framework presented in this paper includes numerous design choices, and many methods can be subsumed under it. This extensive configurability provides possibilities for improved efficiency but also makes it more complex to set up. More confined research is required in order to learn the effect of different configurations and the optimal approaches to various problem settings.

Data partitioning is an extremely active research area with several black box algorithms being proposed by various research groups (e.g. Kucukelbir et al., 2016; Hasenclever et al., 2015; Bardenet et al., 2015). We are sure that different methods will be more effective in different problems. The present paper has two roles: we review the steps that are needed to keep EP algorithms numerically stable, and we are suggesting a general approach, inspired by EP, for approaching data partitioning problems. We anticipate that great progress could be made by using message passing to regularize existing algorithms.

Acknowledgements

We thank David Blei and Ole Winther for helpful comments, and the U.S. National Science Foundation, Institute for Education Sciences, Office of Naval Research, Moore and Sloan Foundations, and Academy of Finland (grant 298742) for partial support of this research. We also acknowledge the computational resources provided by the Aalto Science-IT project.

References

Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Anoop Korattikara Balan, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 181–189, 2014.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. *arXiv.org*, 2015.

Simon Barthelmé and Nicolas Chopin. Expectation propagation for likelihood-free inference. *Journal of the American Statistical Association*, 109:315–333, 2014.

Michael Betancourt. A general metric for Riemannian manifold Hamiltonian Monte Carlo. In Frank Nielsen and Frédéric Barbaresco, editors, *Geometric Science of Information - First International Conference*, pages 327–334, Berlin, Heidelberg, 2013. Springer.

Michael Betancourt. Adiabatic Monte Carlo. *arXiv preprint arXiv:1405.3489*, 2014.

Taras Bodnar and Arjun K. Gupta. Estimation of the precision matrix of a multivariate elliptically contoured stable distribution. *Statistics*, 45(2):131–142, 2011.

Taras Bodnar, Arjun K. Gupta, and Nestor Parolya. Optimal linear shrinkage estimator for large dimensional precision matrix. *arXiv preprint arXiv:1308.0931*, 2014.

Siddhartha Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90(432):1313–1321, 1995.

Jean-Marie Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2012.

Botond Cseke and Tom Heskes. Approximate marginals in latent Gaussian models. *Journal of Machine Learning Research*, 12:417–454, 2011.

John P. Cunningham, Philipp Hennig, and Simon Lacoste-Julien. Gaussian probabilities and expectation propagation. *arXiv preprint arXiv:1111.6832*, 2011.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le, and Andrew Y. Ng. Large Scale Distributed Deep Networks. In *Neural Information Processing Systems*, pages 1223–1231, 2012.

Guillaume Dehaene and Simon Barthelme. Expectation Propagation in the large-data limit. *arXiv preprint arXiv:1503.08060*, 2015.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society B*, 68(3):411–436, 2006.

Francesca Dominici, Giovanni Parmigiani, Robert L. Wolpert, and Vic Hasselblad. Meta-analysis of migraine headache treatments: Combining information from heterogeneous designs. *Journal of the American Statistical Association*, 94(445):16–28, 1999.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Andrew Gelman, Frederic Bois, and Jiming Jiang. Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association*, 91:1400–1412, 1996.

Andrew Gelman, Aleks Jakulin, Maria Grazia Pittau, and Yu-Sung Su. A weakly informative default prior distribution for logistic and other regression models. *Annals of Applied Statistics*, 2: 1360–1383, 2008.

Andrew Gelman, Bob Carpenter, Michael Betancourt, Marcus Brubaker, and Aki Vehtari. Computationally efficient maximum likelihood, penalized maximum likelihood, and hierarchical modeling using Stan. Technical report, Department of Statistics, Columbia University, 2014.

John Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339, 1989.

Arjun K. Gupta, Tamas Varga, and Taras Bodnar. *Elliptically Contoured Models in Statistics and Portfolio Theory*. Springer-Verlag, New York, 2 edition, 2013.

Erika T. Hamden, David Schiminovich, and Mark Seibert. The diffuse galactic far-ultraviolet sky. *The Astrophysical Journal*, 779(180):15, December 2013.

Leonard Hasenclever, Stefan Webb, Thibaut Lienart, Sebastian Vollmer, Balaji Lakshminarayanan, Charles Blundell, and Yee Whye Teh. Distributed Bayesian learning with stochastic natural-gradient expectation propagation and the posterior server. *arXiv preprint arXiv:1512.09327*, 2015.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernández-Lobato, and Richard E. Turner. Black-box α-divergence minimization. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, 2016.

Tom Heskes and Onno Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, pages 216–223, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

Julian P. T. Higgins and Anne Whitehead. Borrowing strength from external trials in a meta-analysis. *Statistics in Medicine*, 15(24):2733–2749, 1996.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Michael I. Jordan. An introduction to probabilistic graphical models. Technical report, 2003.

Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust Gaussian process regression with a Student-t likelihood. *Journal of Machine Learning Research*, 12:3227–3257, 2011.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference. *arXiv preprint arXiv:1603.00788*, 2016.

Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems*, pages 2323–2331, 2015.

Thomas P. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, Cambridge, MA, USA, 2001a.

Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In J. Breese and D. Koller, editors, *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI-2001)*, pages 362–369. Morgan Kaufmann, San Francisco, Clif., 2001b.

Thomas P. Minka. Power EP. Technical report, Microsoft Research, Cambridge, 2004.

Thomas P. Minka. Divergence measures and message passing. Technical report, Microsoft Research, Cambridge, 2005.

Thomas P. Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, pages 352–359. Morgan Kaufmann, San Francisco, CA, 2002.

Robb J. Muirhead. *Aspects of Multivariate Statistical Theory.* John Wiley & Sons, Hoboken, New Jersey, 2005.

Willie Neiswanger, Chong Wang, and Eric P. Xing. Asymptotically exact, embarrassingly parallel MCMC. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*, pages 623–632, 2014.

Manfred Opper and Ole Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

Manfred Opper and Ole Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6:2177–2204, 2005.

Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3): 241–288, 1986.

Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 33, pages 814–822, 2014.

Rajesh Ranganath, Dustin Tran, and David M. Blei. Hierarchical variational models. In *International Conference on Machine Learning*, 2016.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Jaakko Riihimäki, Pasi Jylänki, and Aki Vehtari. Nested expectation propagation for Gaussian process classification with a multinomial probit likelihood. *Journal of Machine Learning Research*, 14:75–109, 2013.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal statistical Society B*, 71(2):319–392, 2009.

Amadou Sarr and Arjun K. Gupta. Estimation of the precision matrix of multivariate Kotz type model. *Journal of Multivariate Analysis*, 100(4):742–752, 2009.

Anand D. Sarwate, Sergey M. Plis, Jessica A. Turner, Mohammad R. Arbabshirani, and Vince D. Calhoun. Sharing privacy-sensitive access to neuroimaging and genetics data: A review and preliminary validation. *Frontiers in Neuroinformatics*, 8(35), 2014. doi: 10.3389/fninf.2014.00035.

Steven L. Scott, Alexander W. Blocker, and Fernando V. Bonassi. Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 2016. URL http://www.tandfonline.com/doi/full/10.1080/17509653.2016.1142191.

Matthias Seeger. Expectation propagation for exponential families. Technical report, Max Planck Institute for Biological Cybernetics, Tubingen, 2005.

Matthias Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.

Matthias Seeger and Michael I. Jordan. Sparse Gaussian process classification with multiple classes. Technical report, University of California, Berkeley, 2004.

Alexander J. Smola, Vishy Vishwanathan, and Eleazar Eskin. Laplace propagation. In S. Thrun, L.K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing 16*, 2004.

Stan Development Team. *Stan modeling language: User's guide and reference manual*, 2016. Version 2.14.0, http://mc-stan.org/.

Dustin Tran, Rajesh Ranganath, and David M. Blei. The variational Gaussian process. In *International Conference on Learning Representations*, 2016.

Hisayuki Tsukuma and Yoshihiko Konno. On improved estimation of normal precision matrix and discriminant ccoefficients. *Journal of Multivariate Analysis*, 97(7):1477–1500, 2006.

Marcel van Gerven, Botond Cseke, Robert Oostenveld, and Tom Heskes. Bayesian source localization with the multivariate Laplace prior. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing 22*, pages 1901–1909, 2009.

Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. GPstuff: Bayesian modeling with Gaussian processes. *Journal of Machine Learning Research*, 14:1175–1179, 2013.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Pareto smoothed importance sampling. *arXiv:1507.02646*, 2016.

Mattias Villani and Rolf Larsson. The multivariate split normal distribution and asymmetric principal components analysis. *Communications in Statistics—Theory and Methods*, 35(6):1123–1140, 2006.

Matt P. Wand. Fast approximate inference for arbitrarily large semiparametric regression models via message passing. *Journal of the American Statistical Association*, 2017.

Xiangyu Wang and David B. Dunson. Parallelizing MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

John Winn and Christopher M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

Minjie Xu, Balaji Lakshminarayanan, Yee Whye Teh, Jun Zhu, and Bo Zhang. Distributed Bayesian posterior sampling via moment sharing. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3356–3364, 2014.

Onno Zoeter and Tom Heskes. Gaussian quadrature based expectation propagation. In Robert Cowell and Zoubin Ghahramani, editors, *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, volume 10, 2005.

Appendices

A. Distributed parallel algorithms

This section presents detailed algorithms for distributed EP applied in the context of partitioned data. Extended algorithms for hierarchical models and cases where dimension reduction is possible are also presented.

A.1. Distributed parallel message passing algorithm

In this subsection we give a practical algorithm description, which is suitable for implementing the general message passing algorithms discussed in Sections 3 and 4 in a numerically stable manner. We assume that the posterior distribution,

$$p(\theta|y) = Z^{-1} \prod_{k=1}^{K} p(y_k|\theta)p(\theta),$$

is approximated by

$$g(\theta) = Z_{\text{EP}}^{-1} \prod_{k=1}^{K} Z_k^{-1} g(\theta|r_k, Q_k) Z_0^{-1} g(\theta|r_0, Q_0) = \text{N}(\theta|\mu, \Sigma),$$

where the site approximations $g_k(\theta) = Z_k^{-1} g(\theta|r_k, Q_k)$ and the prior $p(\theta) = \text{N}(\theta|\mu_0, \Sigma_0) = Z_0^{-1} g(\theta|r_0, Q_0)$ are written using the following definitions:

$$\begin{aligned} g(\theta|r, Q) &= \exp\Big(-\frac{1}{2}\theta' Q\theta + r'\theta\Big) \\ \Psi(r, Q) = \log \int g(\theta|r, Q)\, \mathrm{d}\theta &= \frac{1}{2}\Big(-\log(|Q/2\pi|) + r' Q^{-1} r\Big). \end{aligned} \tag{3}$$

The natural parameters and the normalization of the prior $p(\theta)$ are given by $r_0 = \Sigma_0^{-1}\mu_0$, $Q_0 = \Sigma_0^{-1}$, and $\log Z_0 = \Psi(r_0, Q_0)$. The natural exponential parameters of the posterior approximation $g(\theta)$ are obtained by multiplying the prior and the site approximations together, which gives

$$Q = \Sigma^{-1} = \sum_{k=1}^{K} Q_k + Q_0 \quad \text{and} \quad r = \Sigma^{-1}\mu = \sum_{k=1}^{K} r_k + r_0.$$

The approximate posterior mean $\mu = Q^{-1}r$ and covariance $\Sigma = Q^{-1}$ can be computed using a Cholesky or eigenvalue decomposition of Q. One possibility to initialize the site approximations is to set them to $g_k(\theta) = 1$ by choosing $r_k = 0$ and $Q_k = 0$ for $k = 1, \ldots, K$, which is equivalent to initializing $g(\theta)$ to the prior, that is, $\mu = \mu_0$ and $\Sigma = \Sigma_0$.

We propose to distribute the cavity and tilted moment computations and the site parameter updates to K different computing units. The posterior update is done in the central computing node in parallel fashion. First, the site updates are initialized to zero as $(\Delta r_k = 0, \Delta Q_k = 0)_{k=1}^K$ and then the following steps are repeated until convergence:

1. In parallel at each node: Compute the updated site parameters with damping level $\delta \in (0, 1]$:

$$\begin{aligned} Q_k^{\text{new}} &= Q_k + \delta \Delta Q_k \\ r_k^{\text{new}} &= r_k + \delta \Delta r_k. \end{aligned}$$

2. At the central node: Compute the natural parameters of $g(\theta)^{\text{new}}$ as

$$Q^{\text{new}} = \sum_{k=1}^{K} Q_k^{\text{new}} + Q_0$$
$$r^{\text{new}} = \sum_{k=1}^{K} r_k^{\text{new}} + r_0.$$

3. In parallel at each node: Determine the cavity distributions $g_{-k}(\theta) = \mathrm{N}(\mu_{-k}, \Sigma_{-k})$ for all $k = 1, \ldots, K$:

$$Q_{-k} = Q^{\text{new}} - \eta Q_k^{\text{new}}$$
$$r_{-k} = r^{\text{new}} - \eta r_k^{\text{new}},$$

   where $\eta \in (0, 1]$.

4. In parallel at each node: If $|Q_{-k}| \leq 0$ for any k, go back to step 1 and decrease δ. Otherwise, accept the new state by setting $r = r^{\text{new}}$, $Q = Q^{\text{new}}$, and $(Q_k = Q_k^{\text{new}}, r_k = r_k^{\text{new}})_{k=1}^{K}$ and continue to step 5.

5. In parallel at each node: determine the natural parameters $r_{\backslash k} = \Sigma_{\backslash k}^{-1}\mu_{\backslash k}$ and $Q_{\backslash k} = \Sigma_{\backslash k}^{-1}$ of the tilted distribution $g_{\backslash k}(z_k)$ using either MCMC or Laplace's method. The tilted distribution is given by

$$\begin{aligned} g_{\backslash k}(\theta) &= Z_{\backslash k}^{-1} p(y_k|\theta)^{\eta} \mathrm{N}(\theta|Q_{-k}^{-1}\mu_{-k}, Q_{-k}^{-1}) \\ &\propto p(y_k|\theta)^{\eta} \exp\left(-\frac{1}{2}\theta' Q_{-k}\theta + r'_{-k}\theta\right), \end{aligned}$$

   which can be efficiently sampled and differentiated using

$$\log g_{\backslash k}(\theta) = \eta \log p(y_k|\theta) - \frac{1}{2}\theta' Q_{-k}\theta + r'_{-k}\theta + \text{const.}$$

   Key properties of the different approximation methods are:

   - MCMC: It is easy to compute $\mu_{\backslash k}$ and $\Sigma_{\backslash k}$ from a set of samples, and $\Sigma_{\backslash k}$ should be symmetric and positive definite if enough samples are used. Various approaches for computing the precision matrix $Q_{\backslash k} = \Sigma_{\backslash k}^{-1}$ are discussed in Section A.2 and in Section 5.4 in the original article.
   - Laplace's method: Gradient-based methods can be used to determine the mode of the tilted distribution efficiently. Once a local mode $\hat{\theta}$ is found, the natural parameters can be computed as

$$\begin{aligned} Q_{\backslash k} &= -\nabla_\theta^2 \log g_{\backslash k}(\theta)|_{\theta=\hat{\theta}} = -\eta \nabla_\theta^2 \log p(y_k|\theta)|_{\theta=\hat{\theta}} + Q_{-k} \\ r_{\backslash k} &= Q_{\backslash k}\hat{\theta}. \end{aligned}$$

     If $\hat{\theta}$ is a local mode, $Q_{\backslash k}$ should be symmetric and positive definite.

6. In parallel at each node: If $|Q_{\backslash k}| > 0$, compute the undamped site parameter updates resulting from the moment consistency conditions $Q_{\backslash k} = Q_{-k} + \eta Q_k^{\text{new}}$ and $r_{\backslash k} = r_{-k} + \eta r_k^{\text{new}}$:

$$\Delta Q_k = Q_k^{\text{new}} - Q_k = \eta^{-1}(Q_{\backslash k} - Q_{-k}) - Q_k$$
$$\Delta r_k = r_k^{\text{new}} - r_k = \eta^{-1}(r_{\backslash k} - r_{-k}) - r_k,$$

   If $|Q_{\backslash k}| \leq 0$, there are at least two options: discard the update by setting $\Delta Q_k = 0$ and $\Delta r_k = 0$, or use the SoftAbs map to improve the conditioning of $Q_{\backslash k}$ and compute the parameter updates with the modified $Q_{\backslash k}$.

   In the latter approach, the natural location parameter of the tilted distribution can be recomputed as $r_{\backslash k} = Q_{\backslash i}\mu_{\backslash k}$ using the original tilted mean $\mu_{\backslash k}$ and the modified covariance matrix $Q_{\backslash k}$, which preserves the tilted mean $\mu_{\backslash k}$ but changes the tilted covariance estimate $\Sigma_{\backslash k}$.

Steps 1–6 are repeated until all the tilted distributions are consistent with the approximate posterior, that is, $r = r_{\backslash k}$ and $Q = Q_{\backslash k}$ for $k = 1, \ldots, K$. Step 4 is done to ensure that the posterior approximation $g(\theta)$ and the cavity distributions $g_{-k}(\theta)$ remain well-defined at all times. Step 4 is potentially time consuming because it involves checking the conditioning of all the cavity precision matrices $\{Q_k\}_{k=1}^K$. A cheaper alternative could be to skip the step and apply more damping in the first place, which we expect should work well if the tilted distribution related to the different data pieces are not very different or multimodal.

A.1.1. Discussion of advantages and limitations

Advantages

- The central node does not need to compute $O(d^3)$ matrix decompositions in step 2. It only needs to sum together the natural site parameters to obtain the posterior approximation in exponential form, and pass this to the individual computing nodes that can make the subtractions to form the cavity parameters. Furthermore, the summation of multiple site update terms can also be parallelized term-wise, $Q = (Q_1 + Q_2) + (Q_3 + Q_4)$, and element-wise, $[Q]_{i,j} = [Q_1]_{i,j} + [Q_2]_{i,j}$.
- The tilted moments can be determined by sampling directly from the unnormalized tilted distributions or by using the Laplace's method. This requires only cheap function and gradient evaluations and can be applied to a wide variety of models.
- After convergence, the final posterior approximation could be formed by mixing the draws from the different tilted distributions because these should be consistent with each other and with $g(\theta)$. This sample-based approximation could also capture potential skewness in $p(\theta|y)$ because it resembles the EP-based marginal improvements described by Cseke and Heskes (2011).

Limitations

- The tilted distribution covariance matrices can be easily computed from the samples, but obtaining the precision matrix efficiently is problematic. Various methods for dealing with this issue are discussed in Section 5.4 of the original article. These methods often involve computing the inverse of the sample covariance or scatter matrix, which as such is a costly and inaccurate operation. However, as discussed in Section A.2, the QR-decomposition can be used here to more efficiently form the Cholesky factor of the matrix directly from the samples.

- Estimating the marginal likelihood is more challenging, because determining the normalization constants $Z_{\setminus k}$ requires multivariate integrations. For example, annealed importance sampling type of approaches could be used if marginal likelihood estimates are required.

  With the Laplace's method, approximating $Z_{\setminus k}$ is straightforward but the quality of the marginal likelihood approximation is not likely to very good with skewed posterior distributions. The Laplace marginal likelihood estimate is not generally well-calibrated with the approximate predictive distributions in terms of hyperparameter estimation. Therefore, it would make sense to integrate over the hyperparameters within the EP framework.

A.2. Inverting the scatter matrix

When using sample based estimates for the tilted distribution moment estimation, one often needs to deal with the inverse of the covariance or scatter matrix. In practice, one wants to form the Cholesky decomposition for it. The naive way would be to calculate the scatter matrix and apply available routines to determine the factorization. However, here QR-decomposition can be used to compute it directly from the samples without ever forming the scatter matrix itself. This makes the process more stable, as forming the scatter matrix squares the condition number.

Consider the samples concatenated as an $n \times d$ matrix D with column means removed so that the scatter matrix becomes $S = D^T D$. In the QR-decomposition $D = QR$, the matrix R corresponds to the upper triangular Cholesky factor of the scatter matrix, although the rows may be negative. Moreover, because the factor Q is not needed, it is possible to compute the QR-decomposition even more efficiently.

A.3. Efficient algorithms when dimension reduction is possible

Here we summarize a version of the EP algorithm of Section A.1 for the special case in which the non-Gaussian likelihood terms $p(y_k|\theta)$ depend on θ only through low-dimensional linearly transformed random variables,

$$z_k = U_k \theta; \tag{4}$$

that is, $p(y_k|\theta) = p(y_k|z_k)$ for each partition k. The posterior distribution is now given by

$$p(\theta|y) = Z^{-1} \prod_{k=1}^{K} p(y_k|U_k\theta)p(\theta),$$

and we approximate it by

$$g(\theta) = Z_{\mathrm{EP}}^{-1} \prod_{k=1}^{K} Z_k^{-1} g(U_k\theta|r_k, Q_k) Z_0^{-1} g(\theta|r_0, Q_0) = \mathrm{N}(\theta|\mu, \Sigma).$$

The natural parameters of $g(\theta)$ are obtained by multiplying the site approximations and the prior which gives

$$Q = \Sigma^{-1} = \sum_{k=1}^{K} U_k Q_k U_k' + Q_0 \quad \text{and} \quad r = \Sigma^{-1}\mu = \sum_{k=1}^{K} U_k r_k + r_0. \tag{5}$$

The approximate posterior mean $\mu = Q^{-1}r$ and covariance $\Sigma = Q^{-1}$ can be computed using a Cholesky or eigenvalue decomposition of Q, or a series of K rank-d updates. One possibility is to initialize the site approximations to $g_k(\theta) = 1$ by setting $r_k = 0$ and $Q_k = 0$ for $k = 1, \ldots, K$, which is equivalent to initializing $g(\theta)$ to the prior, that is, $\mu = \mu_0$ and $\Sigma = \Sigma_0$.

If U_k is a $k \times d$ matrix, then the cavity computations and the site parameter updates require only rank-d matrix computations, and determining the moments of the kth tilted distribution $g_{\backslash k}(\theta)$ requires only d-dimensional numerical integrations. In the following, we outline how this algorithm can be parallelized using m computing units.

We propose to distribute the cavity and tilted moment computations into m different computing units by dividing the model terms into m non-intersecting subsets S_j so that $\bigcup_{j=1}^m S_j = \{1, \dots, K\}$. The posterior updates are done in the central computing node in a parallel fashion. First, the site updates are initialized to zero, $(\Delta r_k = 0, \Delta Q_k = 0)_{k=1}^K$, and then the following steps are repeated until convergence:

1. Distribute the parameters $(r_k, Q_k, U_k)_{i \in S_j}$ and the site parameter updates $(\Delta r_k, \Delta Q_k)_{i=S_j}$ to each computing node $j = 1, \dots, m$ and compute intermediate natural parameters $(\tilde{Q}_j, \tilde{r}_j)_{j=1}^m$ with damping level $\delta \in (0, 1]$:

   (a) Compute the updated site parameters for $i \in S_j$:

   $$Q_k^{\text{new}} = Q_k + \delta \Delta Q_k$$
   $$r_k^{\text{new}} = r_k + \delta \Delta r_k.$$

   (b) Compute the natural parameters of the jth batch:

   $$\tilde{Q}_j = \sum_{i \in S_j} U_k Q_k^{\text{new}} U_k'$$
   $$\tilde{r}_j = \sum_{i \in S_j} U_k r_k^{\text{new}}.$$

2. At the central node, compute the natural parameters of $g(\theta)^{\text{new}}$ as

   $$Q^{\text{new}} = \sum_{j=1}^m \tilde{Q}_j + Q_0$$
   $$r^{\text{new}} = \sum_{j=1}^m \tilde{r}_j + r_0,$$

   and determine the posterior mean $\mu^{\text{new}} = (Q^{\text{new}})^{-1} r^{\text{new}}$ and covariance $\Sigma^{\text{new}} = (Q^{\text{new}})^{-1}$ using a Cholesky or eigenvalue decomposition.

3. If $|Q^{\text{new}}| \leq 0$, go to step 1 and decrease δ. Otherwise, continue to step 4.

4. Distribute μ^{new}, Σ^{new}, and $(r_k^{\text{new}}, Q_k^{\text{new}}, U_k)_{i \in S_j}$ to each computing node $j = 1, \dots, m$, and determine the cavity distributions $g_{-k}(z_k) = \text{N}(m_{-k}, V_{-k})$ of the transformed random variables $z_k = U_k' \theta$ for all $i \in S_j$:

   $$Q_{-k} = V_{-k}^{-1} = V_k^{-1} - \eta Q_k^{\text{new}}$$
   $$r_{-k} = V_{-k}^{-1} m_{-k} = V_k^{-1} m_k - \eta r_k^{\text{new}},$$

   where $m_k = U_k' \mu^{\text{new}}$ and $V_k = U_k' \Sigma^{\text{new}} U_k$ are the moments of the approximate marginal distribution $g(z_k) = \text{N}(m_k, V_k)$, and $\eta \in (0, 1]$.

   Save $c_k = \Psi(r_{-k}, Q_{-k}) - \Psi(V_k^{-1} m_k, V_k^{-1})$ for computing the marginal likelihood as described in Section D.

5. If $|V_{-k}| \leq 0$ for any k, go back to step 1 and decrease δ. Otherwise, accept the new state by setting $r = r^{\text{new}}$, $Q = Q^{\text{new}}$, $\mu = \mu^{\text{new}}$, $\Sigma = \Sigma^{\text{new}}$ and $(Q_k = Q_k^{\text{new}}, r_k = r_k^{\text{new}})_{k=1}^K$ and continue to step 6.

6. Distribute parameters $(m_{-k}, V_{-k}, r_k, Q_k, U_k)_{i \in S_j}$ to each computing node $j = 1, \ldots, m$ and determine the site parameter updates $(\Delta r_k, \Delta Q_k)_{i=S_j}$ using the following steps:

   (a) Compute the moments $Z_{\backslash k}$, $m_{\backslash k} = \mathrm{E}(z_k)$, and $V_{\backslash k} = \mathrm{var}(z_k)$ of the tilted distribution $g_{\backslash k}(z_k)$ (recall that $z_k = U_k'\theta$ as defined in (4)) either analytically or using a numerical quadrature depending on the functional form of the exact site term $p(y_k|U'\theta)$:

   $$g_{\backslash k}(z_k) = Z_{\backslash k}^{-1} p(y_k|U'\theta)^{\eta} \mathrm{N}(z_k|m_{-k}, V_{-k}) \approx \mathrm{N}(z_k|m_{\backslash k}, V_{\backslash k}),$$

   where $Z_{\backslash k} = \int p(y_k|U'\theta)^{\eta} \mathrm{N}(z_k|m_{-k}, V_{-k}) \, \mathrm{d}z_k$. Save $Z_{\backslash k}$ for computing the marginal likelihood as described in Section D.

   (b) If $Z_{\backslash k} > 0$ and $|V_{\backslash k}| > 0$, compute the undamped site parameter updates resulting from the moment consistency conditions $V_{\backslash k}^{-1} = V_{-k}^{-1} + \eta Q_k^{\text{new}}$ and $V_{\backslash k}^{-1} m_{\backslash k} = V_{-k}^{-1} m_{-k} + \eta r_k^{\text{new}}$:

   $$\Delta Q_k = Q_k^{\text{new}} - Q_k = \eta^{-1}(V_{\backslash k}^{-1} - V_{-k}^{-1}) - Q_k$$
   $$\Delta r_k = r_k^{\text{new}} - r_k = \eta^{-1}(V_{\backslash k}^{-1} m_{\backslash k} - V_{-k}^{-1} m_{-k}) - r_k,$$

   If $Z_{\backslash k} \leq 0$ or $|V_{\backslash k}| \leq 0$, discard the update by setting $\Delta Q_k = 0$ and $\Delta r_k = 0$.

Steps 1–6 are repeated until all the tilted distributions are consistent with the approximate posterior, that is, $m_k = m_{\backslash k}$ and $V_k = V_{\backslash k}$ for $k = 1, \ldots, K$. Steps 3 and 5 are done to ensure that the posterior approximation $g(\theta)$ and the cavity distributions $g_{-k}(z_k)$ remain well-defined at all times. In practice, we expect that these numerical stability checks do not require any additional computations if a suitable damping factor is chosen. An additional approach is to stabilize the computations is to apply more damping to site updates with $\Delta Q_k < 0$, because only this kind of precision decreases can lead to negative cavity distributions.

Without the stability checks, the algorithm can be simplified so that fewer parameter transfers between central and the computing nodes are needed per iteration. The algorithm could be further streamlined by doing the posterior updates at steps 1–5 incrementally one batch at a time.

A.3.1. Discussion of advantages and limitations

Advantages

- If U_k is a $d \times 1$ matrix, only one-dimensional integrations are required to determine the site parameters. Furthermore, with certain likelihoods, the conditional moments with respect to some components of z_k are analytical which can be used to lower dimensionality of the required integrations. This goes against the general black-box approach of this paper but could be relevant for difficult special cases.

- The cavity computations and parameter updates are computationally cheap if d is small. In addition, the required computations can be distributed to the different computing nodes in a parallel EP framework.

Limitations

- The model terms need to depend on low-dimensional transformations of the unknowns $z_k = U_k\theta$. For example generalized linear models and Gaussian processes fall in to this category.
- Different types of model or likelihood terms require specific implementations. For example, probit and finite Gaussian mixture likelihoods can be handled analytically whereas Poisson and student-t likelihoods require quadratures. For a black-box implementation we might prefer to use numerical quadrature for all these problems.
- The central node needs to compute the global posterior covariance at step 2, which scales as $O(d^3)$ and can be tedious with a large number of unknowns. Independence assumptions or multilevel designs, as proposed in Section 4 in the original article, can be used to improve the scaling.

A.4. Parallel EP implementation for hierarchical models when approximations are formed also for the local parameters

In this subsection we describe a distributed EP algorithm that uses the hierarchical model structure from Section 4 of the original article for efficient parallel computations in inference problems, where a Gaussian approximation is required also for the local parameters. Such cases arise, for example, when certain data pieces share some of the local parameters but we do not wish to form the potentially high-dimensional joint approximation of ϕ and all the local parameters.

We assume independent Gaussian priors for ϕ and $\alpha_1, \ldots, \alpha_K$ and approximate the posterior distribution

$$p(\alpha, \phi|y) = Z^{-1}\mathrm{N}(\phi|B_0^{-1}b_0, B_0^{-1}) \prod_{k=1}^{K} p(y_k|\alpha_k, \phi)\mathrm{N}(\alpha_k|A_0^{-1}a_0, A_0^{-1})$$

by

$$g(\alpha, \phi) = Z_{\mathrm{EP}}^{-1}Z_0^{-1}g(\phi|b_0, B_0), \prod_{k=1}^{K} Z_k g(\alpha_k, \phi|r_k, Q_k)g(\alpha_k|a_0, A_0), \tag{6}$$

where the site approximations and the prior terms are written using (3), and Z_0 is the normalization term of the joint prior $p(\alpha, \phi)$. To derive the EP updates for the hierarchical model, we divide the site location vector r_k and the site precision matrix Q_k to blocks corresponding to α_k and ϕ as

$$r_k = \begin{bmatrix} a_k \\ b_k \end{bmatrix} \quad \text{and} \quad Q_k = \begin{bmatrix} A_k & C_k \\ C_k' & B_k \end{bmatrix}.$$

The marginal approximation for the shared parameters ϕ can be obtained by integrating over the local parameters α_k in the joint approximation (6) as

$$g(\phi) = N(\mu_\phi, \Sigma_\phi) \propto g(\phi|b_0, B_0) \prod_{k=1}^{K} \int g(\alpha_k, \phi|r_k, Q_k)g(\alpha_k|a_0, A_0)\,\mathrm{d}\alpha_k \propto g(\phi|b, B),$$

where the parameters of $g(\phi)$ are given by

$$\begin{aligned} b = \Sigma_\phi^{-1}\mu_\phi &= \sum_{k=1}^{K}\Big(b_k - C_k'(A_k + A_0)^{-1}(a_k + a_0)\Big) + b_0 \\ B = \Sigma_\phi^{-1} &= \sum_{k=1}^{K}\Big(B_k - C_k'(A_k + A_0)^{-1}C_k\Big) + B_0. \end{aligned} \tag{7}$$

In the EP update related to data piece y_k, we need to consider only the joint marginal approximation of α_k and ϕ, which can be written as

$$g(\alpha_k, \phi) \propto g(\alpha_k, \phi | r_k, Q_k) g_{-k}(\alpha_k, \phi), \tag{8}$$

where the kth cavity distribution is defined as

$$g_{-k}(\alpha_k, \phi) \propto g(\alpha_k | a_0, A_0) g(\phi | b_{-k}, B_{-k}),$$

with natural parameters

$$\begin{aligned} b_{-k} &= \sum_{j \neq i} \Big(b_j - C_j' (A_j + A_0)^{-1} (a_j + a_0) \Big) + b_0 \\ &= b - \Big(b_k - C_k' (A_k + A_0)^{-1} (a_k + a_0) \Big) \\ B_{-k} &= \sum_{j \neq i} \Big(B_j - C_j' (A_j + A_0)^{-1} C_j \Big) + B_0 \\ &= B - \Big(B_k - C_k' (A_k + A_0)^{-1} C_k \Big). \end{aligned}$$

The cavity distribution $g_{-k}(\alpha_k, \phi)$ factorizes between α_k and ϕ, and the marginal cavity of the local parameters α_k depends only on the prior $p(\alpha_k)$. The dependence on the other local parameters is incorporated in the marginal cavity $g_{-k}(\phi) \propto g(\phi | b_{-k}, B_{-k})$. This property results from the factorized prior between ϕ and $\alpha_1, \ldots, \alpha_K$, and it enables computationally efficient lower-dimensional matrix computations. The marginal approximation $g(\alpha_k) = N(\mu_{\alpha_k}, \Sigma_{\alpha_k})$ can be obtained by integrating over ϕ in (8), which gives

$$\begin{aligned} \Sigma_{\alpha_k} &= \Big(A_0 + A_k - C_k (B_{-k} + B_k)^{-1} C_k' \Big)^{-1} \\ \mu_{\alpha_k} &= \Sigma_{\alpha_k} \Big(a_0 + a_k - C_k (B_{-k} + B_k)^{-1} (b_{-k} + b_k) \Big). \end{aligned} \tag{9}$$

The marginal approximations $g(\phi)$ and $\{g(\alpha_k)\}_{k=1}^K$ can be computed efficiently without actually forming the potentially high-dimensional joint approximation $g(\alpha_1, \ldots, \alpha_K, \phi)$. After convergence, we can summarize the coefficients and compute the predictions for each group $k = 1, \ldots, K$ using the marginal distributions (7) and (9).

Approximations (7) and (9) can be determined by first initializing the site parameters and the parameter updates to zero, that is $(a_k = 0, b_k = 0, A_k = 0, B_k = 0, C_k = 0)_{k=1}^K$ and $(\Delta a_k = 0, \Delta b_k = 0, \Delta A_k = 0, \Delta B_k = 0, \Delta C_k = 0)_{k=1}^K$, and then iterating the following steps until convergence:

1. Distribute the current site parameters $(a_k, A_k, b_k, B_k, C_k)$ together with the parameter updates $(\Delta a_k, \Delta A_k, \Delta b_k, \Delta B_k, \Delta C_k)$ to the corresponding computing node $k = 1, \ldots, K$, and compute new parameter values with damping level $\delta \in (0, 1]$:

$$\begin{aligned} a_k^{\text{new}} &= a_k + \delta \Delta a_k & \qquad b_k^{\text{new}} &= a_k + \delta \Delta a_k \\ A_k^{\text{new}} &= A_k + \delta \Delta A_k & \qquad B_k^{\text{new}} &= B_k + \delta \Delta B_k \\ C_k^{\text{new}} &= C_k + \delta \Delta C_k. \end{aligned}$$

   Compute also auxiliary variables $V_k = (A_k^{\text{new}} + A_0)^{-1}$ using for example K parallel Cholesky decompositions. If $|V_k| \leq 0$, i.e. the Cholesky decomposition fails, decrease δ and recompute the updates. Otherwise, compute auxiliary parameters

$$\begin{aligned} \tilde{b}_k &= b_k^{\text{new}} - (C_k^{\text{new}})' V_k (a_k^{\text{new}} + a_0) \\ \tilde{B}_k &= B_k^{\text{new}} - (C_k^{\text{new}})' V_k C_k^{\text{new}}. \end{aligned}$$

2. At the central node, compute the natural parameters of $g(\phi)^{\text{new}} = \mathrm{N}\left(\mu_\phi^{\text{new}}, \Sigma_\phi^{\text{new}}\right)$ as

$$b^{\text{new}} = (\Sigma_\phi^{\text{new}})^{-1}\mu_\phi^{\text{new}} = \sum_{k=1}^{K} \tilde{b}_k + b_0$$
$$B^{\text{new}} = (\Sigma_\phi^{\text{new}})^{-1} = \sum_{k=1}^{K} \tilde{B}_k + B_0. \tag{10}$$

3. Distribute parameters $(b^{\text{new}}, B^{\text{new}}, b_k^{\text{new}}, B_k^{\text{new}})$ to the respective computing nodes $k = 1, \ldots, K$, and determine the parameters of the cavity distributions:

$$g_{-k}(\alpha_k, \phi) = Z_{-k}^{-1} g(\alpha_k | a_{-k}, A_{-k}) g(\phi | b_{-k}, B_{-k}),$$

where $Z_{-k} = \Psi(a_{-k}, A_{-k}) + \Psi(b_{-k}, B_{-k})$ and

$$a_{-k} = a_0 \qquad b_{-k} = b^{\text{new}} - \tilde{b}_k$$
$$A_{-k} = A_0 \qquad B_{-k} = B^{\text{new}} - \tilde{B}_k.$$

4. If $|B_{-k}| \leq 0$ for any k, go back to step 1 and decrease δ. Another option is to skip updates for sites $\{k, |B_{-k}| \leq 0\}$ but ill-conditioned cavity distributions and approximate covariance matrices may still emerge at subsequent iterations.

   Otherwise, accept the new state by setting $b = b^{\text{new}}$, $B = B^{\text{new}}$, and $(a_k = a_k^{\text{new}}, A_k = A_k^{\text{new}}, b_k = b_k^{\text{new}}, B_k = B_k^{\text{new}}, C_k = C_k^{\text{new}})_{k=1}^K$ and continue to the next step. Save the normalization terms $\log Z_{-k}$ for computing the marginal likelihood as described in Section D.

5. Distribute the parameters $(a_{-k}, A_{-k}, b_{-k}, B_{-k})$ to the corresponding computing nodes $k = 1, \ldots, K$ and determine the site parameter updates by the following steps:

   (a) Determine the normalization term $Z_{\backslash k}$ and the moments $\mu_{\backslash k} = \mathrm{E}(\alpha_k, \phi)$ and $\Sigma_{\backslash k} = \mathrm{cov}(\alpha_k, \phi)$ of the tilted distribution $g_{\backslash k}(\alpha_k, \phi)$ using either an inner EP algorithm or MCMC depending on the functional form of the likelihood term $p(y_k | \alpha_k, \phi)$:

$$\begin{aligned} g_{\backslash k}(\alpha_k, \phi) &= Z_{\backslash k}^{-1} p(y_k | \alpha_k, \phi) Z_{-k}^{-1} g(\alpha_k | a_{-k}, A_{-k}) g(\phi | b_{-k}, B_{-k}) \\ &\approx \mathrm{N}(\alpha_k, \phi | \mu_{\backslash k}, \Sigma_{\backslash k}), \end{aligned}$$

   where $Z_{\backslash k} = \int p(y_k | \alpha_k, \phi) g_{-k}(\alpha_k, \phi) \, \mathrm{d}\alpha_k \, \mathrm{d}\phi$. For the site updates only the natural parameters of the tilted distribution need to be determined:

$$r_{\backslash k} = \Sigma_{\backslash k}^{-1} \mu_{\backslash k} = \begin{bmatrix} a_{\backslash k} \\ b_{\backslash k} \end{bmatrix} \qquad Q_{\backslash k} = \Sigma_{\backslash k}^{-1} = \begin{bmatrix} A_{\backslash k} & C_{\backslash k} \\ (C_{\backslash k})' & B_{\backslash k} \end{bmatrix}.$$

   (b) If $Z_{\backslash k} > 0$ and $|Q_{\backslash k}| > 0$, compute the undamped site parameter updates resulting from the moment consistency conditions $r_{\backslash k} = r_{-k} + r_k^{\text{new}}$ and $Q_{\backslash k} = Q_{-k} + Q_k^{\text{new}}$:

$$\Delta a_k = a_{\backslash k} - a_{-k} - a_k \qquad \Delta b_k = b_{\backslash k} - b_{-k} - b_k$$
$$\Delta A_k = A_{\backslash k} - A_{-k} - A_k \qquad \Delta B_k = B_{\backslash k} - B_{-k} - B_k$$
$$\Delta C_k = C_{\backslash k} - C_k.$$

Save the following quantity for computing the marginal likelihood as described in Section D:

$$c_k = \log Z_{\backslash k} - \Psi\left(\begin{bmatrix} a_{-k} + a_k \\ b_{-k} + b_k \end{bmatrix}, \begin{bmatrix} A_{-k} + A_k & C_k \\ C_k' & B_{-k} + B_k \end{bmatrix}\right).$$

If $Z_{\backslash k} \leq 0$ or $|Q_{\backslash k}| \leq 0$, discard the update by setting $\Delta a_k = 0, \Delta b_k = 0, \Delta A_k = 0, \Delta B_k = 0$, and $\Delta C_k = 0$ for that particular data piece k.

A.5. Determining tilted moments using inner EP approximations when dimension reduction is possible

We can use an inner EP algorithm to determine the natural parameters $r_{\backslash k}$ and $Q_{\backslash k}$ of the tilted distributions $g_{\backslash k}(\alpha_k, \phi)$ in step 5a of the hierarchical EP algorithm in Appendix A.4, if the likelihood terms related to each data piece can be factored into simple terms that depend only on low-dimensional linearly transformed random variables $z_{k,j} = U_{k,j}(\alpha_k, \phi)$, that is,

$$p(y_k|\alpha_k, \phi) = \prod_{j=1}^{n_k} p(y_{k,j}|U_{k,j}(\alpha_k, \phi)),$$

where n_k is the number of observations in batch k. The EP algorithm description with dimension reduction from Section A.3 can be readily applied for this purpose. Since the tilted moment approximations in step 5a of the hierarchical algorithm are already run in parallel at the different computing nodes, the parallelization steps can be excluded when used to from the inner EP approximations.

We can also derive closed-form solutions for the parameters $(a_k, b_k, A_k, B_k, C_k)_{k=1}^K$ of the approximation (6) in terms of the site parameters of the inner EP algorithms. First, we write the approximation to the tilted distributions as

$$\begin{aligned} g_{\backslash k}(\alpha_k, \phi) &= Z_{\backslash k}^{-1} Z_{-k}^{-1} p(y_k|\alpha_k, \phi) g(\alpha_k|a_{-k}, A_{-k}) g(\phi|b_{-k}, B_{-k}) \\ &\approx Z_{\backslash k}^{-1} Z_{-k}^{-1} \prod_{j=1}^{n_k} Z_{k,j} g(U_{k,j}(\alpha_k, \phi|\tilde{r}_{k,j}, \tilde{Q}_{k,j}) g(\alpha_k|a_{-k}, A_{-k}) g(\phi|b_{-k}, B_{-k}), \end{aligned}$$

where $\tilde{r}_{k,j}$ and $\tilde{Q}_{k,j}$ are the site parameters of the inner EP approximation. If we write the transformation as $U_{k,j}(\alpha_k, \phi) = u_{k,j}\alpha_k + v_{k,j}\phi$, where $U_{k,j} = (u_{k,j}, v_{k,j})$, we can write the outer EP parameters as

$$\begin{aligned} a_k &= a_{\backslash k} - a_{-k} = \sum_j u_{k,j}\tilde{r}_{k,j} & b_k &= b_{\backslash k} - b_{-k} = \sum_j v_{k,j}\tilde{r}_{k,j} \\ A_k &= A_{\backslash k} - A_{-k} = \sum_j u_{k,j}\tilde{Q}_{k,j}u_{k,j}' & B_k &= B_{\backslash k} - B_{-k} = \sum_j v_{k,j}\tilde{Q}_{k,j}v_{k,j}' \\ C_k &= C_{\backslash k} = \sum_j u_{k,j}\tilde{Q}_{k,j}v_{k,j}'. \end{aligned}$$

For example, in case of a linear model, $u_{k,j}$ are the input variables associated with the local coefficients α_k, and $v_{k,j}$ are the input variables corresponding to the shared coefficients ϕ.

With this representation, we can interpret the hierarchical EP algorithm with K inner EP approximations also as a single algorithm with site parameters $\tilde{r}_{k,j}$ and $\tilde{Q}_{k,j}$ that are updated in parallel fashion for K groups of site terms. After each successful update at step 4 we store only parameters $\tilde{r}_{k,j}$ and $\tilde{Q}_{k,j}$, and we can also equivalently apply damping directly to these parameters

at step 1. In fact, it is more efficient to initialize each tilted moment approximation at step 5a to the inner EP parameters from the previous iteration instead of starting from a zero initialization. This framework is similar to the nested EP algorithm for the multinomial probit model described by Riihimäki et al. (2013). However, if applied to the potentially high-dimensional hierarchical setting, the computationally benefits become more evident compared with the GP classification case studied in that paper.

B. Implementation in Stan

We implement our experiments with Python, R and Stan. Whether using point estimation (for Laplace approximations) or HMC (for simulations from the tilted distribution), we write a Stan program that includes one portion of the likelihood and an expression for the cavity distribution. We then run this model repeatedly with different inputs for each subset k. This ensures that only one part of the likelihood gets computed at a time by the separate processes, but it does have the cost that separate Stan code is needed to implement the message passing computations. In future software development, we would like to be able to take an existing Stan model and merely overlay a factorization so that the message passing algorithm could be applied directly to the model.

We use Stan to compute the tilted distribution moments in each distributed site. Currently we perform the other steps in Python or R (sample code available at `https://github.com/gelman/ep-stan`). In parallel EP, we pass the normal approximations g_k back and forth between the master node and the K separate nodes.

Currently, Stan performs adaptation each time it runs, but future versions should allow restarting from the previous state, which will speed up computation substantially when the algorithm starts to converge. We should also be able to approximate the expectations more efficiently using importance sampling.

C. The computational opportunity of parallel message passing algorithms

We have claimed that message passing algorithms offer computational gains for large inference problems by splitting the data into pieces and performing inference on each of those pieces in parallel, occasionally sharing information between the pieces. Here we detail those benefits specifically.

Consider the simple, non-hierarchical implementation in Section 3 in the original article with a multivariate normal approximating family. We assume that we have $K+1$ parallel processing units: one central processor that maintains the global posterior approximation $g(\theta)$ and K worker units on which inference can be computed on each of the K factors of the likelihood. Furthermore, we assume a network transmission cost of c per parameter. Let N be the number of data points and let D be the number of parameters, that is, the length of the vector θ. Finally, we define $h(n, d)$ as the computational cost of approximating the tilted distribution (a task which may, for example, be performed by running MCMC to perform simulations) with n data points and d parameters.

Each step of the algorithm then incurs the following costs:

1. **Partitioning.** This loading and caching step will in general have immaterial cost.

2. **Initialization.** The central unit initializes the site approximations $g_k(\theta)$, which by construction are multivariate normal. In the general case each of the K sites will require $D+D(D+1)/2$ scalar quantities corresponding to the mean and covariance. Thus the central unit bears the initialization cost of $O(KD^2)$.

3. **EP iteration.** Let m be the number of iterations over all K sites. Empirically m is typically a manageable quantity; however, numerical instabilities tend to increase this number. In parallel EP, damped updates are often used to avoid oscillation (van Gerven et al., 2009).

   (a) Computing the cavity distribution. Owing to our multivariate normal approximating family, this step involves only simple rank D matrix operation per site, costing $O(D^3)$ (with a small constant; see Cunningham et al., 2011). One key choice is whether to perform this step at the central unit or worker units. If we compute the cavity distributions at each worker unit, the central unit must first transmit the full posterior to all K worker units, costing $O(cND^2)$ for cost c per network operation. In parallel, the cavity computations then incur total cost of $O(D^3)$. On the other hand, small D implies central cavity computations are preferred, requiring $O(KD^3)$ to construct K cavity distributions centrally, with a smaller subsequent distribution cost of $O(cKD^2)$. Accordingly, the total cost per EP iteration is $O(\min\{cND^2 + D^3, cKD^2 + KD^3\})$. We presume any computation constant is much smaller than the network transmission constant c, and thus in the small D regime, this step should be borne by the central unit, a choice strengthened by the presumed burden of step 3b on the worker units.

   (b) Fitting an updated local approximation $g_k(\theta)$. We must estimate $O(D^2)$ parameters. More critical in the large data setting is the cost of computing the log-likelihoods. In the best case, for example if the likelihoods belong to the same exponential family, we need only calculate a statistic on the data, with cost $O(N/K)$. In some rare cases the desired moment calculation will be analytically tractable, which results in a minimum cost of $O(D^2 + N/K)$. Absent analytical moments, we might choose a modal approximation (e.g. Laplace propagation), which may typically incur a $O(D^3)$ term. More common still, MCMC or another quadrature approach over the D-dimensional site parameter space will be more costly still: $h(N/K, D) > D^2$. Furthermore, a more complicated likelihood than the exponential family—especially a multimodal $p(y_k|\theta)$ such as a mixture likelihood—will significantly influence numerical integration. Accordingly, in that common case, this step costs $O(h(N/K, D)) \gg O(D^2 + N/K)$. Critically, our setup parallelizes this computation, and thus the factor K is absent.

   (c) Return the updated $g_k(\theta)$ to the central processor. This cost repeats the cost and consideration of step 3a.

   (d) Update the global approximation $g(\theta)$. In usual parallel EP, $g(\theta)$ is updated once after all site updates. However, if the cost h of approximating the posterior distribution is variable across worker units (for example, in an MCMC scheme), the central unit could update $g(\theta)$ whenever possible or according to a schedule.

Considering only the dominating terms, across all these steps and the m EP iterations, we have the total cost of our parallel message passing algorithm:

$$O\Big(m\Big(\min\{cND^2 + D^3, cKD^2 + KD^3\} + h(N/K, D)\Big)\Big).$$

This cost contains a term due to Gaussian operations and a term due to parallelized tilted approximations. By comparison, consider first the cost of a non-parallel EP algorithm:

$$O\Big(m\Big(ND^3 + Nh(N/K, D)\Big)\Big).$$

Second, consider the cost of full numerical quadrature with no partitioning:

$$O(h(N, D)).$$

With these three expressions, we can immediately see the computational benefits of our scheme. In many cases, numerical integration will be by far the most costly operation, and will depend superlinearly on its arguments. Thus, our parallel message passing scheme will dominate. As the total data size N grows large, our scheme becomes essential. When data is particularly big (e.g. $N \approx 10^9$), our scheme will dominate even in the rare case that h is its minimal $O(D^2 + N/K)$ (see step 3b above).

D. Marginal likelihood

Although not the focus of this work, we mention in passing that EP also offers as no extra cost an approximation of the marginal likelihood, $p(y) = \int p_0(\theta)p(y|\theta)\,\mathrm{d}\theta$. This quantity is often used in model choice.

To this end, associate to each approximating site g_k a constant Z_k, and write the global approximation as:

$$g(\theta) = p_0(\theta) \prod_{k=1}^{K} \frac{1}{Z_k} g_k(\theta).$$

Consider the Gaussian case, for the sake of simplicity, so that $g_k(\theta) = e^{-\frac{1}{2}\theta' Q_k \theta + r_k'\theta}$, under natural parameterization, and denote by $\Psi(r_k, Q_k)$ the corresponding normalizing constant:

$$\psi(r_k, Q_k) = \int e^{-\frac{1}{2}\theta' Q_k \theta + r_k'\theta}\,\mathrm{d}\theta = \frac{1}{2}(-\log|Q_k/2\pi| + r_k' Q_k r_k).$$

Simple calculations (Seeger, 2005) then lead to following formula for the update of Z_k at site k,

$$\log(Z_k) = \log(Z_{\backslash k}) - \Psi(r, Q) + \Psi(r_{-k}, Q_{-k}),$$

where $Z_{\backslash k}$ is the normalizing constant of the tilted distribution $g_{\backslash k}$, (r, Q) is the natural parameter of g, $r = \sum_{k=1}^{K} r_k$, $Q = \sum_{k=1}^{K} Q_k$, $r_{-k} = \sum_{j \neq i} r_j$, and $Q_{-k} = \sum_{j \neq i} Q_j$. In the deterministic approaches we have discussed for approximating moments of $g_{\backslash k}$, it is straightforward to obtain an approximation of the normalizing constant; when simulation is used, some extra efforts may be required, as in Chib (1995).

Finally, after completion of EP one should return the following quantity,

$$\sum_{k=1}^{K} \log(Z_k) + \Psi(r, Q) - \Psi(r_0, Q_0),$$

as the EP approximation of $\log p(y)$, where (r_0, Q_0) is the natural parameter of the prior.