Expectation Propagation as a Way of Life: A Framework for Bayesian Inference on Partitioned Data

Aki Vehtari   AKI.VEHTARI@AALTO.FI
Department of Computer Science
Aalto University
00076 Aalto, Finland

Andrew Gelman   GELMAN@STAT.COLUMBIA.EDU
Departments of Statistics and Political Science
Columbia University
New York, NY 10027, USA

Tuomas Sivula   TUOMAS.SIVULA@AALTO.FI

Pasi Jylänki   PASI.JYLANKI@GMAIL.COM

Dustin Tran   DUSTINVIETTRAN@GMAIL.COM

Swupnil Sahai   SWUPNIL@STAT.COLUMBIA.EDU

Paul Blomstedt   PAUL.BLOMSTEDT@AALTO.FI

John P. Cunningham   JPC2181@COLUMBIA.EDU

David Schiminovich   DS@ASTRO.COLUMBIA.EDU

Christian P. Robert   XIAN@CEREMADE.DAUPHINE.FR

Editor: Manfred Opper

Abstract

A common divide-and-conquer approach for Bayesian computation with big data is to partition the data, perform local inference for each piece separately, and combine the results to obtain a global posterior approximation. While being conceptually and computationally appealing, this method involves the problematic need to also split the prior for the local inferences; these weakened priors may not provide enough regularization for each separate computation, thus eliminating one of the key advantages of Bayesian methods. To resolve this dilemma while still retaining the generalizability of the underlying local inference method, we apply the idea of expectation propagation (EP) as a framework for distributed Bayesian inference. The central idea is to iteratively update approximations to the local likelihoods given the state of the other approximations and the prior.

The present paper has two roles: we review the steps that are needed to keep EP algorithms numerically stable, and we suggest a general approach, inspired by EP, for approaching data partitioning problems in a way that achieves the computational benefits of parallelism while allowing each local update to make use of relevant information from the other sites. In addition, we demonstrate how the method can be applied in a hierarchical context to make use of partitioning of both data and parameters. The paper describes a general algorithmic framework, rather than a specific algorithm, and presents an example implementation for it.

1. Introduction

Expectation propagation (EP) is a fast and parallelizable method of distributional approximation via data partitioning. Since its introduction by Opper and Winther (2000) and Minka (2001b), EP has been an important Bayesian computational method 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. The main theme of our paper is to use EPs message passing technique as a framework for distributed Bayesian inference. The problem is treated in the general setting of combining inferences on data partitioned into disjoint subsets. This setting can be motivated from two complementary views of distributed computing, 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 data sets, 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 motivation for distributing the inference may be to decrease run time or deal with memory limitations. 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 should 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, the motivation for distributed inference may come from the local nature of the data and the model. 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, 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 across separate analyses, enabling more informed decisions both globally and for 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 any single processor having complete access to all the local data or model.

Extracting the core principles behind EP motivates a general framework for passing information between inferences on partitioned data. In classical EP, the data are typically partitioned pointwise, with the approximating density fully factorized. When data are partitioned into bigger subsets, the same idea can be used in a more versatile manner. Here the *cavity distribution*, which approximates the effect of inferences from all other $K - 1$ data partitions, can be used as a prior in the inference step for individual partitions. Xu et al. (2014) apply the EP algorithm in a distributed setting by using MCMC for performing inference for the separate sites. A posterior server maintains a global approximation and iteratively issues updates for distributed sites. At each site, a local inference is carried out and the obtained posterior sample is used to form moment estimates for updating the local approximation. The site updates are propagated back to the posterior server and the global approximation is updated. Motivated by an earlier preprint version of the present paper, Hernández-Lobato and Hernández-Lobato (2016) apply a similar distributed approach for Gaussian process classification. Instead of MCMC, they apply nested EP updates in each site and use the same distributed cluster for updating the hyperparameters between iterations.

EP is not in general guaranteed to converge, which motivates an alternative direction of algorithmic development that, instead of local updates, applies various energy optimization techniques directly to the related objective function. Motivated by Xu et al. (2014) and our earlier preprint version of the present paper (Gelman et al., 2014b), Hasenclever et al. (2017) develop such a distributed algorithm called stochastic natural gradient expectation propagation (SNEP), which also uses MCMC for the site inferences. Similar to the methods presented by Heskes and Zoeter (2002) and Opper and Winther (2005), they implement a convergent double-loop optimization algorithm, which has the same optima as power EP (Minka, 2004). This method, however, introduces some additional computational complexities compared to the local updating power EP method. In the case both methods converge, they are expected to produce similar results. If power EP fails to converge, it is also possible to switch on the fly to the double-loop to ensure convergence as demonstrated by Jylänki et al. (2011).

The work by Xu et al. (2014) and Hasenclever et al. (2017) are particular implementations of the distributed inference framework discussed in this paper. Compared to these works, we consider the method in a more general setting and introduce some further considerations. In addition to just EP, power EP, or SNEP, the framework can be generalized to implement other message passing techniques, all sharing the same idea of using a cavity distribution to iteratively share information between the distributed sites. We do not argue that any particular implementation of the framework would be in general better than the other. Different implementations have different properties that are suitable and desirable in different situations. Apart from a couple of small scale simulated experiments, we do not provide exhaustive comparison of the performance of different implementations of the distributed EP framework against each other.

We consider our method in an applied context and discuss and analyze various algorithmic considerations related to it. For example, we discuss the implementation of sample based moment estimates for EP updates. Xu et al. (2014) presented an estimate for the required

parameters and conjectured that these would be unbiased, but this is not generally true. Using experiments similar to those of Xu et al. (2014), we also study how the number of partitions affects the resulting approximation. Compared to conventional fully factored EP, partitioning the data into bigger subsets may yield better approximations but longer computation times. We additionally show that dividing the data into smaller pieces tends to make the posterior variance estimates worse while maintaining the accuracy of the mean estimates. This is a known property of variational inference but has not been so well understood with EP, although it has been recognized earlier in some form by Cunningham et al. (2011) and Cseke et al. (2013).

We discuss the idea in a generalized 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, 2017), leveraging its sample-based inferences for the individual partitions. We test the implementation in two experiments, in which we inspect the behaviour of EP in the context of the generalized framework.

The remainder of the paper proceeds as follows. We 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

The distributed inference framework presented in this paper is based on the expectation propagation algorithm. In this section, we present EP along with the more generalized idea of a message passing algorithm. Later in Section 2.2, we present various considerations, extensions, and related methods.

2.1. Basic Algorithm

Expectation propagation (EP) is an iterative algorithm in which a target density $f(\theta)$ is approximated by a density $g(\theta)$ from some specified parametric family. First introduced by Opper and Winther (2000) and shortly after generalized by Minka (2001a,b), 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 factor $k = 0$ to the prior $p(\theta)$ and factors 1 through K as the likelihood for the data partitioned into K parts $p(y_k|\theta)$ that are independent given the model parameters. A message passing algorithm works by iteratively approximating $f(\theta)$ with a density $g(\theta)$ which admits the same factorization,

$$g(\theta) \propto \prod_{k=0}^{K} g_k(\theta), \tag{1}$$

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*, and the approximating distribution $g(\theta)$ is referred to as the *global approximation*.

At each iteration of the algorithm, and for $k = 0, \ldots, K$, 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 (and 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 $k \in \{0, 1, \ldots, K\}$ (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 the 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

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

where $D(\cdot\|\cdot)$ corresponds to some divergence measure. In our definition, the algorithm can more freely implement any approximation method, which does not necessarily minimize any divergence.

The global approximation $g(\theta)$ and the site approximations $g_k(\theta)$ are restricted to be in a selected exponential family, such as the multivariate normal. As step 2b is usually defined, the site approximation $g_k(\theta)$ is updated so that the resulting Kullback-Leibler divergence $\mathrm{KL}\big(f_k(\theta)g_{-k}(\theta)\big\|g_k(\theta)g_{-k}(\theta)\big)$ is minimized.

2.2. Further Considerations

The exponential family restriction in EP makes the algorithm efficient: any product and division between these 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 would require 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; Cseke 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 typically 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, beyond the standard choices in the EP literature. If $f_k(\theta)$ has the same form as g then the contribution of that term can be computed exactly and there is no need for the corresponding site approximation term $g_k(\theta)$. For example, if the prior $f_0(\theta)$ and approximating distribution g are both multivariate normals, then only tilted distributions $k = 1, \ldots, K$ need to be computed.

Even if EP minimizes local KL-divergence in the scope of each site, it will not in general minimize the KL-divergence from the target density to the global approximation $\mathrm{KL}(f(\theta)\|g(\theta))$. Furthermore, there is no general guarantee of convergence for EP. However, for models with log-concave factors f_k and initialization to the prior distribution, the algorithm has proven successful in many applications. Various studies have been made to assess the behaviour of EP. Dehaene and Barthelmé (2015) present bounds for the approximate error. Dehaene and Barthelmé (2018) inspect the method in the large data limit and show that it is asymptotically exact but it may diverge if initialized poorly. Dehaene (2016) relate EP to other better understood methods and show that it is equivalent to performing gradient descent on a smoothed energy landscape.

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 Barthelmé (2018) and Li et al. (2015) present a modified EP method in which sites share the same approximate factor $g_{\mathrm{site}}(\theta)$; considering the prior $p(\theta)$ as a constant site with index 0, and setting all the other site approximations $g_k(\theta) = g_{\mathrm{site}}(\theta), k = 1, 2, \ldots, K$, the global approximation becomes $g(\theta) \propto p(\theta)g_{\mathrm{site}}(\theta)^K$. While making the algorithm more memory

efficient, it has been shown for certain applications that the method works almost as well as the original EP.

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 to its own site:

$$p(\theta|y) \propto p(\theta) \prod_{k=1}^{K} p(y_k|\theta), \tag{2}$$

where each term $p(y_k|\theta)$ with respective data partition $y_k = [y_{k,1}, y_{k,2}, \ldots, y_{k,n_k}]$ is approximated with site approximation $g_k(\theta)$. The algorithm can be run in a distributed setting consisting of a central node and site nodes. The schema is illustrated in Figure 1. The central node stores the current global approximation $g(\theta)$ and controls the messaging for the sites, while each site node stores the corresponding partition of the data y_k and the current site approximation $g_k(\theta)$. The central node initiates the updates by sending the natural parameters of the current global approximation $g(\theta)$ to the sites. Given this information, the sites update the respective site approximations $g_k(\theta)$ and send back the change in the natural parameters of the site distribution:

$$
\begin{aligned}
g_{-k}(\theta) &\propto g(\theta)/g_k(\theta) && \text{subtraction of natural parameters} \\
g_{\backslash k}(\theta) &\propto p(y_k|\theta)g_{-k}(\theta) && \text{MCMC sampling} \\
g^{\text{new}}(\theta) &\approx g_{\backslash k}(\theta) && \text{parameter sample estimates} \\
\Delta g_k(\theta) &\propto g^{\text{new}}(\theta)/g(\theta) && \text{subtraction of natural parameters.}
\end{aligned}
$$

The central node then receives the differences and aggregates these to update the global approximation by adding in the received parameter changes:

$$
g^{\text{new}}(\theta) \propto g(\theta)\Delta g_k(\theta) \qquad\qquad \text{sum of natural parameters.}
$$

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). We present the algorithm in more detail in Appendix A.1.

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 should be smaller as the number of sites decreases.

In EP, as mentioned in Section 2, 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
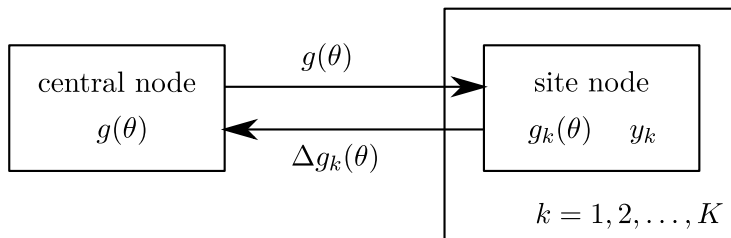
Figure 1: The EP framework for partitioned data. The central node stores the current parameters for the global approximation $g(\theta)$. Each site node $k = 1, 2, \ldots, K$ stores the current parameters for the site approximation $g_k(\theta)$ and the assigned partition of the data y_k. The central node sends the parameters of $g(\theta)$ to the site nodes. In parallel, the site nodes update $g_k(\theta)$ and send back the difference in the parameters.

conveniently inferred by estimating the tilted distribution moments, for example using MCMC. 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 2 illustrates this advantage with a conceptual example, showing how the inference for each site factor $f_k(\theta)$ can be concentrated in a region where all site factors overlap. Figure 3 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.

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 local 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.
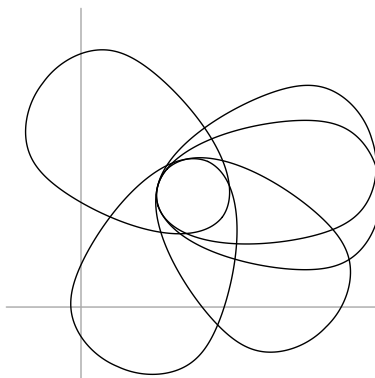
Figure 2: Sketch illustrating the benefits of message passing in Bayesian computation. In this simple example, the parameter space θ has two dimensions, and the data have been split into five pieces. Each oval represents a contour of the likelihood $p(y_k|\theta)$ provided by a single partition of the 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.
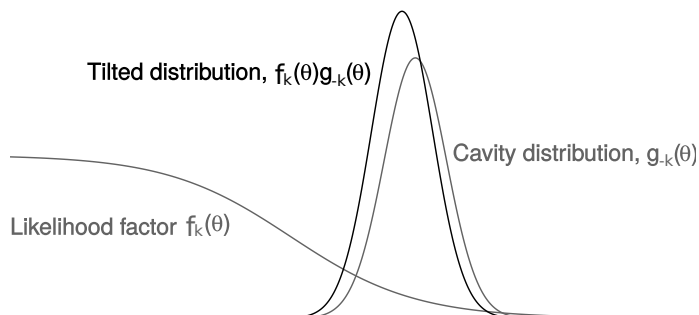

Figure 3: Example of a step of an EP algorithm in a simple one-dimensional example, illustrating the stability of the computation even when part of the likelihood is far from Gaussian. When performing inference on the likelihood factor $p(y_k|\theta)$, the algorithm uses the cavity distribution $g_{-k}(\theta)$ as a prior.
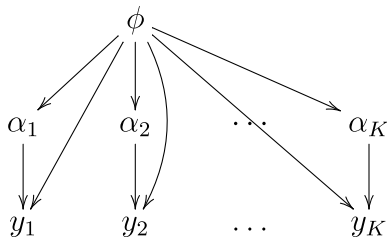
Figure 4: Model structure for the hierarchical EP algorithm. In each site k, inference is based on the local model, $p(y_k|\alpha_k, \phi)p(\alpha_k|\phi)$, multiplied by the cavity distribution $g_{-k}(\phi)$. Computation on this tilted posterior gives a distributional approximation on (α_k, ϕ) or simulation 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 $1/K$.

4.1. 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 of the data model and hyperparameters as well. This structure is displayed in Figure 4. Each data piece y_k is assigned to one site with its own local model $p(y_k|\alpha_k, \phi)p(\alpha_k|\phi)$. The posterior distribution is

$$
\begin{aligned}
p(\phi, \alpha|y) &\propto p(\phi, \alpha)p(y|\phi, \alpha) = p(\phi)p(\alpha|\phi)p(y|\phi, \alpha) \\
&= p(\phi) \prod_{k=1}^{K} p(y_k|\alpha_k, \phi)p(\alpha_k|\phi),
\end{aligned}
\tag{3}
$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$.

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 $p(\phi|y)$ is straightforward. Marginalizing the joint posterior distribution in (3) gives

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

which is approximated by

$$
p(\phi|y) \approx g(\phi) = p(\phi) \prod_{k=1}^{K} g_k(\phi).
$$

10

Given the cavity distribution $g_{-k}(\phi)$, each site k approximates the tilted distribution

$$g_{\backslash k}(\phi) \propto \int g_{-k}(\phi) p(y_k|\alpha_k, \phi) p(\alpha_k|\phi) \, \mathrm{d}\alpha_k \tag{4}$$

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 divide the problem into $K = 300$ sites with 10 groups each, we have reduced a problem with 300 000 data points and 6 020 parameters to 300 parallel iterated problems with 1000 data points and 40 parameters (20 local and 20 shared parameters) each.

4.2. Posterior Inference for the Other Parameters

In large-dimensional hierarchical scenarios, the full joint posterior distribution is not typically needed. If all that is required are the marginal posterior distributions for each α_k separately, we can take these directly from the corresponding tilted distribution inferences from the last iteration. The marginal posterior distribution for local parameter α_k can be obtained from the joint distribution in (3) by

$$\begin{aligned} p(\alpha_k|y) &= \int \int_{\alpha \backslash \alpha_k} p(\phi, \alpha|y) \, \mathrm{d}\alpha_p \, \mathrm{d}\phi \\ &\propto \int p(\phi) p(y_k|\alpha_k, \phi) p(\alpha_k|\phi) \prod_{p \neq k} \int p(y_p|\alpha_p, \phi) p(\alpha_p|\phi) \, \mathrm{d}\alpha_p \, \mathrm{d}\phi. \end{aligned}$$

Assuming the EP algorithm has converged, this can be approximated:

$$p(\alpha_k|y) \approx \int g_{-k}(\phi) p(y_k|\alpha_k, \phi) p(\alpha_k|\phi) \, \mathrm{d}\phi,$$

which is the same as the tilted distribution in (4) but marginalized over ϕ instead of α_k. If, for example, a sample-based method is used for the tilted distribution inference in EP, one can easily just store the local parameter samples in the last iteration to form the marginal posterior distribution for them.

If the joint posterior distribution of all the parameters is required, one can approximate it using the obtained EP approximation $g(\phi)$ for the marginal posterior distribution of the shared parameters:

$$p(\phi, \alpha|y) = p(\phi|y) p(\alpha|\phi, y) = p(\phi|y) \prod_{k=1}^{K} p(\alpha_k|\phi, y_k) \approx g(\phi) \prod_{k=1}^{K} p(\alpha_k|\phi, y_k).$$

To get simulation draws from this, one can first take some number of draws from $g(\phi)$, and then, for each draw, run K parallel MCMC inferences 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 MCMC inferences—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.** 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 most of the memory accesses can be made using fast but small cache memory.

- **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 5.4.

- **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 $g_k(\theta) = \mathrm{N}(0, \frac{1}{K}A^2 I)$, where A is some large value (for example, if the elements of θ are roughly scaled to be of order 1, we might set $A = 10$).

- **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 do not need to be proper distributions. However, improper site distributions may lead to improper global approximations or tilted distributions in subsequent iterations, which is a problem. Various methods for dealing with this issue are discussed in Section 5.3.

In the following sections, we address some of these issues in detail, namely, how to approximate the tilted distribution and how to handle potential numerical instabilities in the algorithms. The methods and aspects discussed in this section cover multiple different implementations for the distributed EP method. Different methods may work in different situations and, as in statistical analysis in general, one has to choose one that suits the problem. In this paper, we present all the prominent approaches in a high level while focusing in one implementation, where the tilted distribution inference is carried out by sampling. With this approach, the inference can be carried out conveniently with probabilistic programming tools, which provides substantial generalizability.

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 $g_{\setminus k}(\theta)$. When applied to Gaussian processes, this approach has the particular advantage that the tilted distribution $g_{\setminus k}(\theta)$ 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.

5.1.1. Mode-based Tilted Approximations

The simplest message passing algorithms construct an approximation of the tilted distribution around its mode at each step. As Figure 3 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 $g_{\setminus k}(\theta)$, with covariance matrix equal to the inverse of the negative Hessian of $\log g_{\setminus k}$ 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 method has been shown to work well in many problems (e.g., Rue et al., 2009).

The presence of the cavity distribution as a prior (as illustrated in Figure 3) 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., 2014a) 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., 2019) or central composite design integration (Rue et al., 2009) to get a better approximation of the moments of the tilted distribution to used in constructing g_k.

5.1.2. 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 from the tilted distribution to the global approximation, $\mathrm{KL}(g_{\setminus k}(\theta)||g^{\mathrm{new}}(\theta))$. 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 the reverse KL divergence from the global approximation to the tilted distribution, $\mathrm{KL}(g^{\mathrm{new}}(\theta)||g_{\setminus k}(\theta))$. 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 $g(\theta) = p(\theta) \prod_{k=1}^{K} g_k(\theta)$.

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.

5.1.3. Energy Optimization

The EP algorithm, like message passing algorithms in general, is not guaranteed to converge. It is possible, however, to define an objective function whose stationary points corresponds to a fixed point for the EP algorithm. In its general form, the problem can be formulated as an optimization for the free energy corresponding to the negative logarithm of the intractable normalizer of the global approximation in Equation (1) (Opper and Winther, 2005). Appendix F illustrates such a formulation in more detail. Heskes et al. (2005) presents an unifying analysis of the correspondence of various different formulations of the same objective, and Dehaene (2016) relates EP to using a gradient descent on a smoothed energy landscape. Various energy optimization methods, for which convergence is guaranteed, can be applied to directly find analogous approximations. Heskes and Zoeter (2002) show a simulated example where EP fails to converge but a double-loop optimization algorithm is successful. While optimizing similar objective functions and possibly admitting similar distributed local updating frameworks, these algorithms are often slower.

Based on the original EP min-max optimization problem reviewed in Appendix F, Opper and Winther (2005) derived a convergent double-loop optimization algorithm called expectation consistent approximate inference (EC). Recently Hasenclever et al. (2017) presented a similar but faster double-loop optimization algorithm called stochastic natural gradient expectation propagation (SNEP), that shares the same optimum as power EP, and which admits a similar distributed framework as the local updating scheme discussed in this paper. They show that, instead of the natural parameter space, SNEP can be seen as a mean parameter space version of the damped EP update. In the case of convergence, both methods are expected to produce similar results. We briefly compare the methods in a simulated experiment in Appendix D with simulation based site inferences. These experiments show that moment matching can reach convergence faster but may suffer from larger variability. SNEP can be slower and behave chaotically when far from convergence but tends to have smaller variability when reaching stable progression and eventually convergence.

It is also possible to apply moment matching in early iterations in order to have good initial progression and switch to SNEP for more stable convergence after some iterations. Jyländi et al. (2011) present similar ideas in a conventional EP setting in order to ensure convergence. However, convergence problems might also indicate that the approximating family matches poorly with the exact posterior (Minka, 2001b; Jyländi et al., 2011). Thus it could be beneficial to first consider some alternative approximating families.

Black-box α-divergence minimization (Hernández-Lobato et al., 2016) (BB-α) is another example of an energy optimizing algorithm with a tunable α-divergence measure and automatic differentiation. Similar to the stochastic EP method by Li et al. (2015) discussed in Section 2.2, the BB-α method features shared local approximations, which makes it more memory efficient. In distributed settings, however, each site unit needs to store a copy of the shared site distribution anyway. Thus no memory is saved by tying up the factors in this setting.

5.1.4. 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 exponential family prior (typically multivariate normal on parameters that if necessary have been transformed to an unbounded scale) that comes from the cavity distribution.

As with methods such as stochastic variational inference (Hoffman et al., 2013) which take steps based on stochastic estimates, the properties of the estimator affect the convergence properties of the EP algorithm. One way to study convergence is to inspect the expectation of the state of the algorithm at the fixed point of conventional analytic EP. As discussed in Section 2, in the EP update step the KL divergence from the new global approximation to the tilted distribution $\mathrm{KL}(g_{\setminus k}(\theta)||g(\theta))$ is minimized by matching the moments. With a simulation-based method, the expectation of the new global approximation moments in step 2b should then match with the tilted distribution moments. When working with the normal approximation, we would use the unbiased estimates of the mean and covariance of the tilted distribution, which are easily obtained from the simulated sample. Using this estimator would not result in the least possible expected KL divergence in general, however. In addition, in the algorithm, these parameters are ultimately needed in natural form, and estimating them is a complex task in general. This problem is discussed in more detail in Appendix A.4. If needed, 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, a Laplace approximation for skewed distributions.

With sample based estimates, there is a tradeoff between computation time and precision of the estimates. In the local bottom-up view of a distributed inference problem, the time taken for the separate inferences is not crucial. Thus it is appropriate to apply MCMC with suitably large sample sizes for such problems.

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 $\theta_{t,k}^s$, $s = 1, \ldots, S_{t,k}$ from the tilted distribution $g_{\backslash k}^t$, possibly with weights $w_{t,k}^s$; take $w_{t,k}^s \equiv 1$ for an unweighted sample. To progress to node $k + 1$, reweight these simulations as: $w_{t,k+1}^s = w_{t,k}^s g_{\backslash(k+1)}^t (\theta_{t,k}^s) / g_{\backslash k}(\theta_{t,k}^s)$. If the vector of new weights has an effective sample size,

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

that 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 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.

5.1.5. NESTED EP

In a hierarchical setting, the model can be fit using the nested EP approach (Riihimäki et al., 2013; Hernández-Lobato and Hernández-Lobato, 2016), 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

As mentioned in Section 2, although the EP algorithm iteratively minimizes the KL divergences from the tilted distributions to their corresponding approximations, it does not minimize the KL divergence from the target density to the global approximation. 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. Minka and Lafferty (2002) proposes to set $\delta = 1/K$ as a safe rule. 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 damping closer to 1 and decrease it gradually with the iterations without affecting the resulting approximation. In our experiments, by comparing the resulting approximation to a known target, we found out that in the first iteration, $\delta = 0.5$ often resulted in good progression, regardless of the number of sites K. In the following iterations, we obtained good results by decreasing damping gradually to $\delta = 1/K$ in K iterations.

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, and this step can be automated.

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.

Problems with the tilted distribution can arise when many of the site approximations become improper. Constraining the sites to proper distributions (perhaps with the exception of the initial site approximations) can 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 will also be positive definite.

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 only slightly deviates from positive definite, it is justified to use the latter approach 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.

5.4. 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, $g_{\backslash k} = g \frac{p_k}{g_k}$, the tilted distribution;

- For any k_1, k_2, $g_{\backslash k_1, k_2} = g \frac{p_{k_1} p_{k_2}}{g_{k_1} g_{k_2}}$, 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 exact 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

As discussed in Section 5, the distributed EP framework can be applied to problems in various ways. In this section, we implement an algorithm using MCMC for tilted distribution inference, demonstrating in two hierarchical examples: a simulated logistic regression problem and a mixture model applied to astronomy data. More details of the experiments can be found in Appendix B.

The objective of these experiments is to demonstrate the EP framework as a convenient method for distributing inference carried out by general probabilistic programming tools. These experiments do not serve as a thorough examination of the principles of EP in itself or as an exhaustive comparison between competitive distributed inference algorithms.

6.1. Simulated Hierarchical Logistic Regression

We demonstrate the distributed EP algorithm with a simulated hierarchical logistic regression problem, a typical case in statistical analysis. We inspect the behavior of the method when increasing the number of partitions, which is expected to speed up the inference but decrease the approximation accuracy. We compare to consensus Monte Carlo (Scott et al., 2016), an alternative distributed sampling method. Non-sampling based methods are not used as a comparison. In particular, related but non-distributable conventional optimization based variational inference is not analyzed here. We consider full non-distributed MCMC approximation as a reference method. The aim of the experiment is to show that the method is applicable and that it can outperform consensus Monte Carlo (Scott et al., 2016).

In the context of distributed computing, the constructed problem is small with 64 groups and 1280 observations in total. When comparing to the non-distributed inference, the gains in the computational efficiency should be greater with bigger problems. The time complexity of the distributed algorithm is mostly determined by the MCMC sampling in the local sites. Thus the expected complexity simplifies to $\mathrm{O}(h(n/K, d_\phi))$, where n is the total number of observations, K is the number of sites, d_ϕ is the dimensionality of the shared parameters, and $h(n/K, d_\phi)$ indicates the complexity of sampling the local model with n/K observations and dimensionality d_ϕ. More detailed analysis of the computational complexity is presented in Appendix C. In general, memory efficiency and limitations must also be
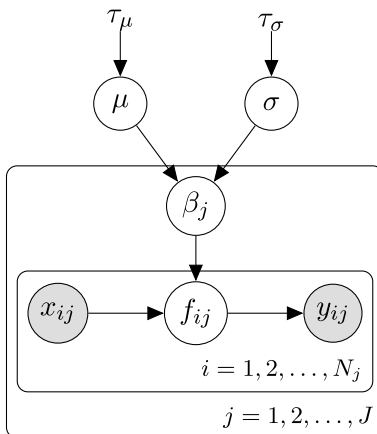
Figure 5: A graphical model representation of the experimented hierarchical logistic regression problem. Indexing $j = 1, 2, \ldots, J$ corresponds to hierarchical groups and $i = 1, 2, \ldots, n_j$ corresponds to observations in group j. Gray nodes represent observed variables and white nodes represent unobserved latent variables. Variables without circles denote fixed priors.

taken into consideration, as the data set might not fit in the memory in the first place and then it would need to be partitioned.

The problem has not been chosen here with the expectation that it would be particularly easy to approximate with the method. On the contrary, it can be seen from the results that unlike in the non-hierarchical logistic regression, where EP is known to perform well, the hierarchical problem hard as EP tends to underestimate the variance when there are many sites and strong posterior dependencies (Cunningham et al., 2011; Cseke et al., 2013).

The model we shall fit is

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

where

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

for all dimensions $d = 0, 1, \ldots, D$, groups $j = 1, 2, \ldots, J$, and observations $i = 1, 2, \ldots, n_j$. The observed data have D features. The first coefficient β_0 corresponds to the intercept; correspondingly, the first element in the data vector $x_{i,j}$ is a column of 1's. The shared parameters inferred with EP are $\phi = (\mu, \log \sigma)$. Figure 5 shows the structure of the model.

The simulated problem is constructed with a $D = 16$ dimensional explanatory variable resulting in a total of $d_\phi = 2(D + 1) = 34$ shared parameters. The number of hierarchical

groups is $J = 64$ and the number of data points per group is $n_j = 20$ for all $j = 1, \ldots, J$, resulting in a total of $N = 1280$ data points. 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 $\mathrm{logit}^{-1}(\beta_j^{\mathrm{T}} x_{ij})$ gets values near 0 and 1 with low but not too low frequency. This ensures that the problem is neither too easy nor too hard. We present the details of the regularization in Appendix B.3.

Following the hierarchical EP algorithm description in Section 4, we run experiments partitioning the data into $K = 2, 4, 8, 16, 32, 64$ sites, using uniform distributions as initial site approximations. Distributing the problem further into $K > 64$ sites, and ultimately to $K = \sum_{j=1}^{J} n_j$ sites corresponding to the conventional fully factored EP, would require that the local parameters be included in the global EP approximation, thus loosing the advantage of the hierarchical setting. This drastic increase in the shared parameter space would often make the approach inapplicable and thus we omit these experiments here.

Our implementation uses Python for the message passing framework and the Stan probabilistic modeling language (Stan Development Team, 2017) for MCMC sampling from the tilted distribution. The tilted distribution moments are estimated in natural form with (5) and (6) from the obtained sample. Each parallel MCMC run has 8 chains of length 200, in which the first halves of the chains are discarded as warmup. In our implementation, the warmup period, during which sampling parameters are learned, is performed in every iteration of EP. It would be possible, however, to adopt the state of the sampler from previous iteration to speed up the process. As discussed before in Section 5.2, we apply gradually decreasing damping factor δ. In our experiment, the following setup produced good results; in the first iteration, $\delta = 0.5$ and it decays exponentially towards $\min(1/K, 0.2)$ while reaching 90 % decay at iteration K.

We compare the results from the distributed EP approximations to a distributed consensus Monte Carlo approximation (Scott et al., 2016) and undistributed full MCMC approximation with varying sample size. In the consensus method, the data is split analogously to $K = 2, 4, 8, 16, 32, 64$ partitions and the prior is respectively fractioned to $p(\theta)^{1/K}$ in each separate inference. All of the obtained results are compared to a target full MCMC approximation with 8 chains of length 10 000, in which the first halves of the chains are 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 20 data points per group, each of the local posterior distributions would be wide, as sketched in Figure 2. The message passing framework, in which at each step the cavity distribution is used as a prior, keeps computations more stable and focused.

Figure 6 illustrates the progression of the experiment for each run. In this experiment, EP reached better accuracy than consensus MC with all K. As shown in Appendix B.2, the difference in the accuracy between EP and consensus MC becomes bigger, if the parameter correlations in the KL divergence measure are ignored. With small K, EP was able to reach accuracy comparable to the full sampling. Figure 7 compares the final obtained approximation accuracy between EP and consensus method with varying K. In both of these methods, the final approximation quality is better with fewer sites but more sites provide opportunities for faster convergence and reduced memory usage per unit in parallelized setting. Figure 8 shows a comparison between posterior mean and standard deviation
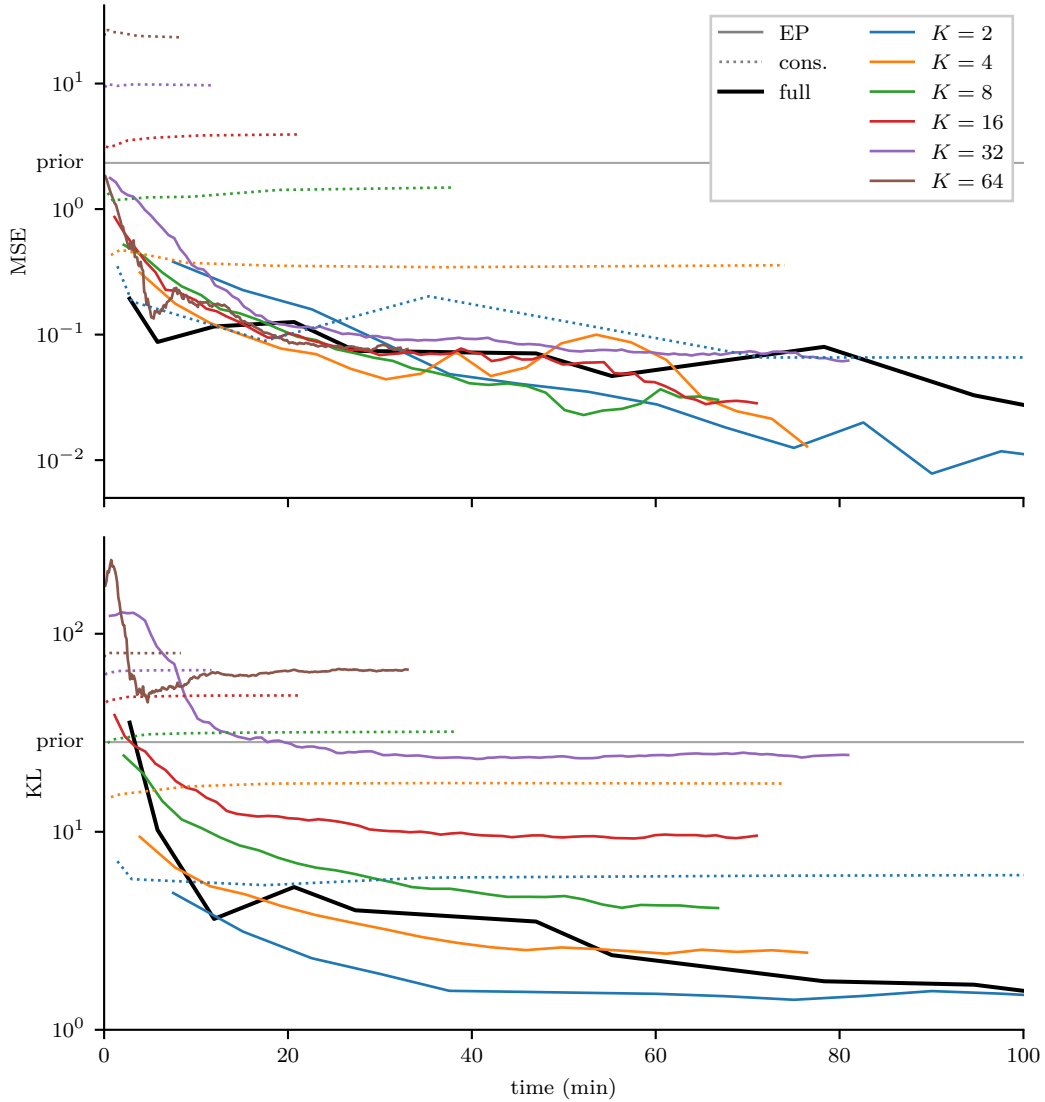
Figure 6: MSE of the mean and approximate KL divergence from the target distribution to the resulting posterior approximation as a function of the elapsed sampling time. Three methods are compared: full MCMC, distributed EP, and distributed consensus MC. For EP (solid lines) and consensus Monte Carlo (dotted lines), line colors indicate the number of partitions K. The y-axis is in the logarithmic scale. Unsurprisingly, the final accuracy declines as the number of partitions increases. In all partitionings, EP outperforms consensus MC, and with small K, it reaches comparable accuracy to the full MCMC approximation. The sampling time comparison is tentative, as the EP implementation could be further optimized by reusing sampling parameters in consecutive iterations. In addition to the time efficiency, the reduced memory usage per distributed unit, gained by increasing the number of partitions K, would also be a concern for large problems.
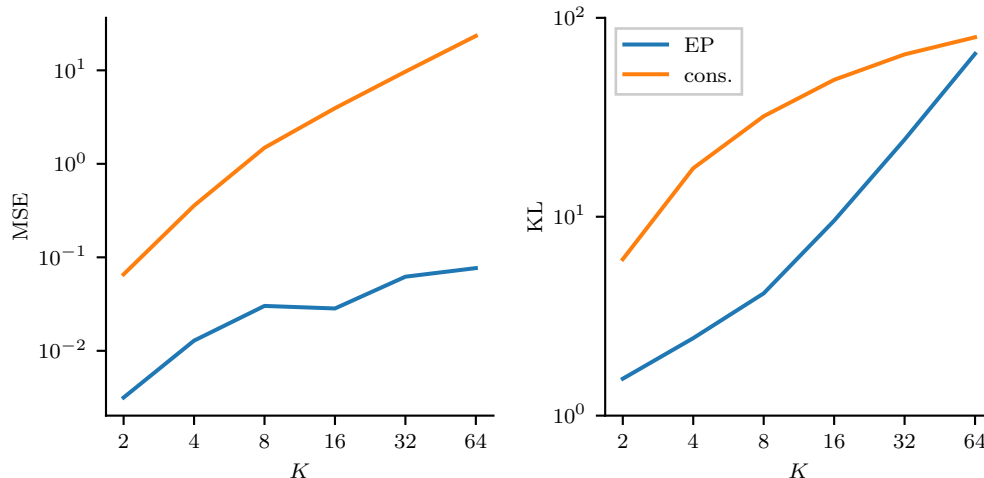
Figure 7: The resulting MSE and KL divergence with distributed EP and consensus Monte Carlo as a function of the number of partitions. EP reached better results in all cases. The y-axis is in the base-10 logarithmic scale and the x-axis is in the base-2 logarithmic scale.

between the distributed EP approximation and the target 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.

As discussed before in the start of this section, it can be seen from Figure 8 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; Cseke et al., 2013). However, unlike with the consensus MC method, the mean is well approximated with distributed EP even with a high number of partitions, as can be seen from Figure 6.

Although not the focus of this experiment, in Figure 6, we assess the time efficiency of the method by inspecting the performance indicator as a function of the time spent in the sampling parts of the code. By this, we can compare the methods in an even manner by neglecting the implementation-specific factor. Each of the methods uses the same Stan implementation for the sampling. In our experiments, the time spent in other parts of the code is minuscule compared to the sampling time; even in the most extreme case of $K = 64$, the time spent in non-sampling parts of the code was only 0.2% of the total time spent. However, as various aspects of the problem affect the computational efficiency, our general time comparison is tentative. For example, it should be possible to improve the sampling time in EP by adopting the sampling parameters from previous EP iterations. We further discuss the computational complexity of the method in Appendix C.
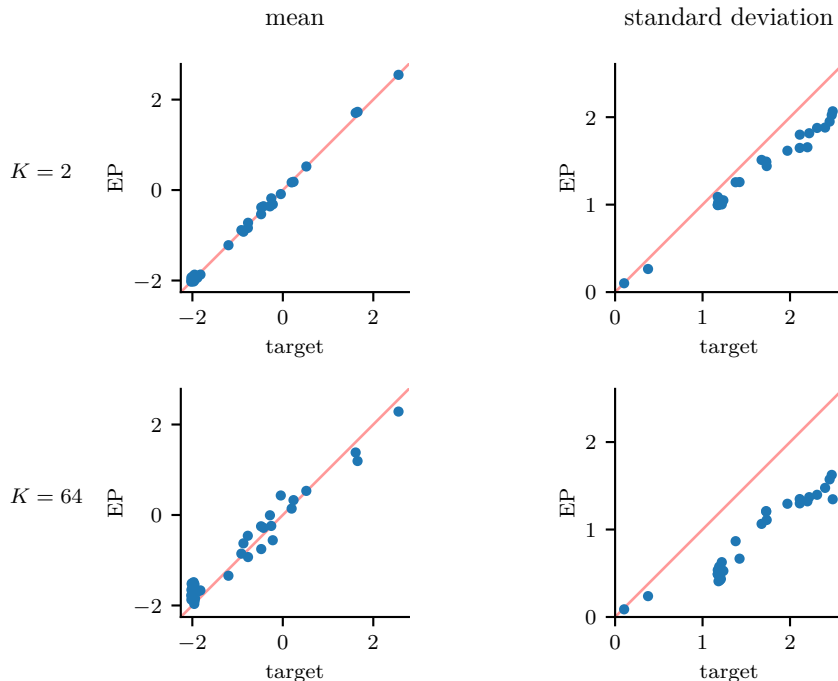
Figure 8: Pointwise comparison of the posterior mean and standard deviation of the target and the final distributed EP approximation when the groups are distributed into $K = 2$ (top row) and $K = 64$ (bottom row) sites. Each dot corresponds to one of the 34 shared parameters. The red diagonal line shows the points of equivalence. It can be seen that in this experiment, while accurately finding the mean, EP systematically underestimates the variance.

6.2. Hierarchical Mixture Model Applied to an Astronomy Problem

We next apply the distributed EP algorithm to a problem in astronomy, where the goal is to model the nonlinear relationship between diffuse galactic far ultraviolet radiation (FUV) and 100-μm infrared emission (i100) in various sectors of the observable universe. The data were collected from the Galaxy Evolution Explorer telescope. An approximate linear relationship has been found between FUV and i100 below i100 values of 8 MJy sr^{-1} (Hamden et al., 2013). Here we attempt to model the nonlinear relationship across the entire span of i100 values, allowing the curves to vary spatially. Sahai (2018) discusses the experiment in more detail and also presents some additional simulated experiments using the same method.

Figure 9 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 nonlinear 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.
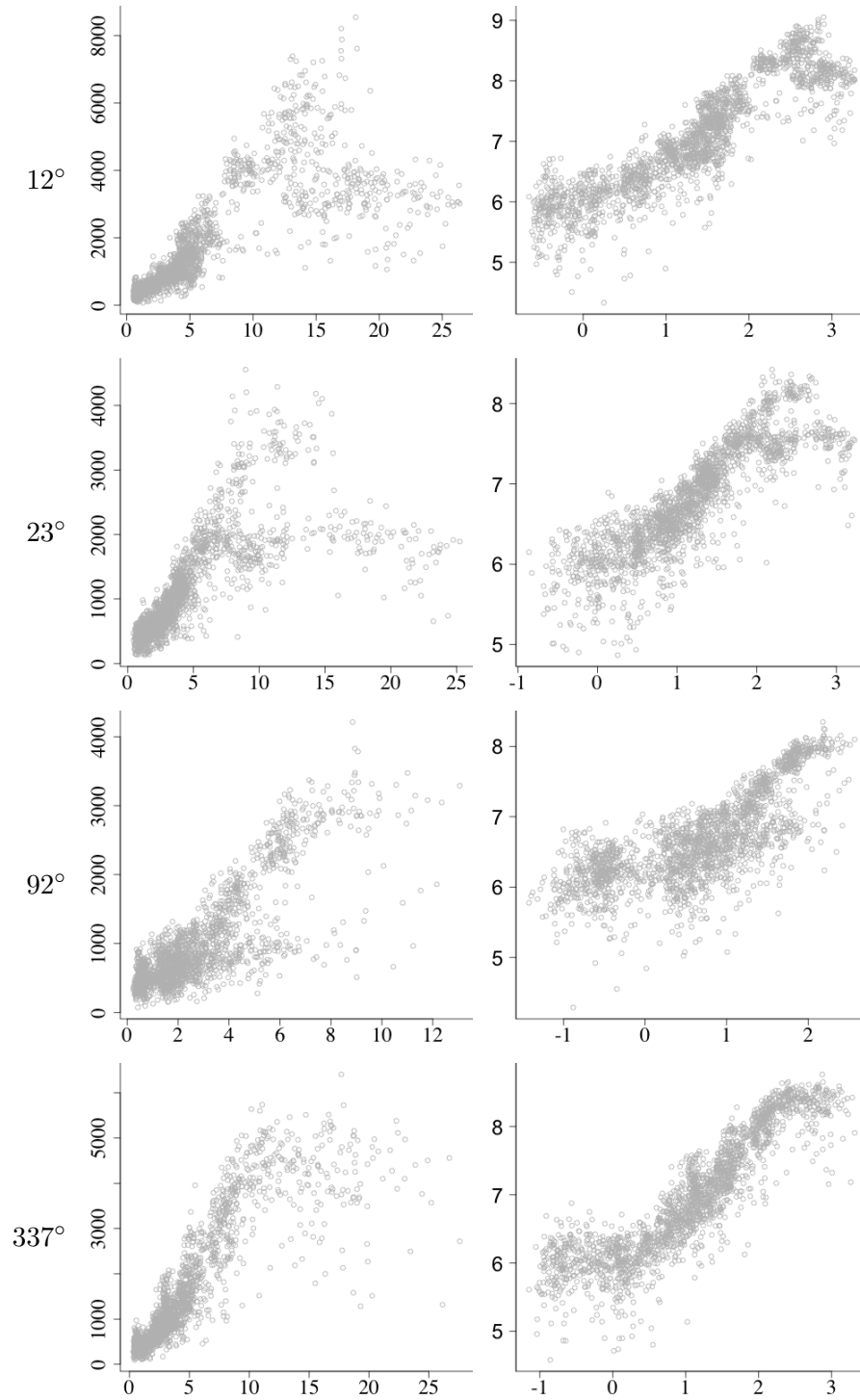
25

Figure 9: Scatterplots of far 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).

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

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

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

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

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:

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

for all groups $j = 1, 2, \ldots, J$, and observation $i = 1, 2, \ldots, n_j$. The model is illustrated graphically in Figure 10.

Hence the 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 $9J = 3\,240$ 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 $N = 720\,000$ observations.

When dividing the longitudinal degrees into distinct hierarchical groups, the relative angular distance between groups is ignored; nearby groups are considered equally dependent
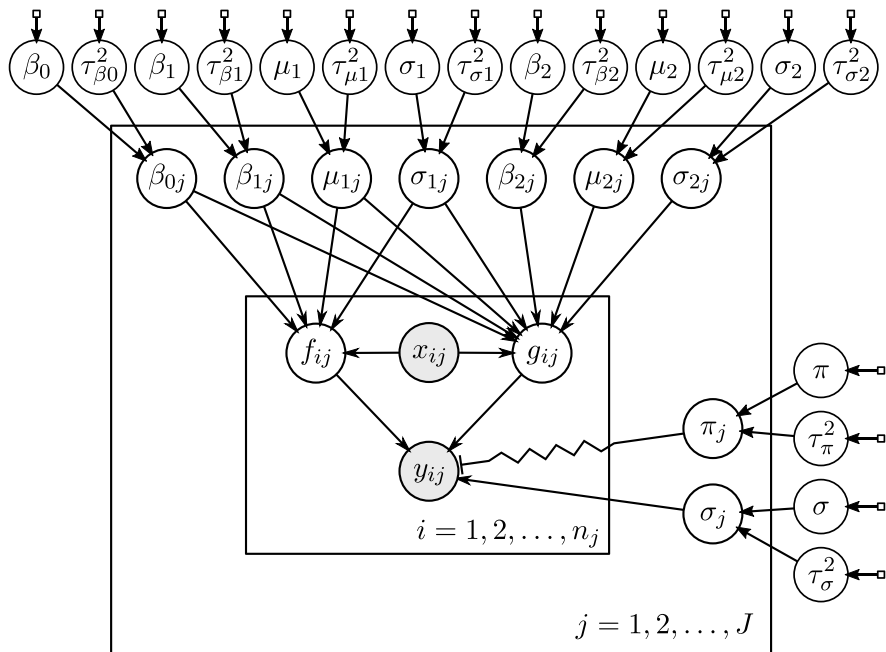
Figure 10: Graphical representation of the astronomy model. Circles represent random variables and boxes represent fixed parameters. Grayed circles are observed. The zig-zag line indicates that π_j functions as a selector between f_{ij} and g_{ij}. The labels for the fixed prior parameters are omitted for clarity.

as far away ones. This is often an issue with divide-and-conquer algorithms when the data have spatial or temporal structure. Increasing the number of partitions ignores more information but also increases computational efficiency. In addition, one must pay attention to local coherence in the groupings. We find that applying this model for the problem is reasonable, and it also serves as an example for hierarchical nonlinear regressions more generally. Sahai (2018) discusses the matter in more detail.

Our implementation uses R for the message passing framework and the Stan probabilistic modeling language (Stan Development Team, 2017) 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 using (5) and (6). Each parallel MCMC run has 4 chains with 1000 iterations each, of which half are discarded as warmup. We use a constant damping factor of $\delta = 0.1$ 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 each, of which half are discarded as warmup.

Figure 11 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
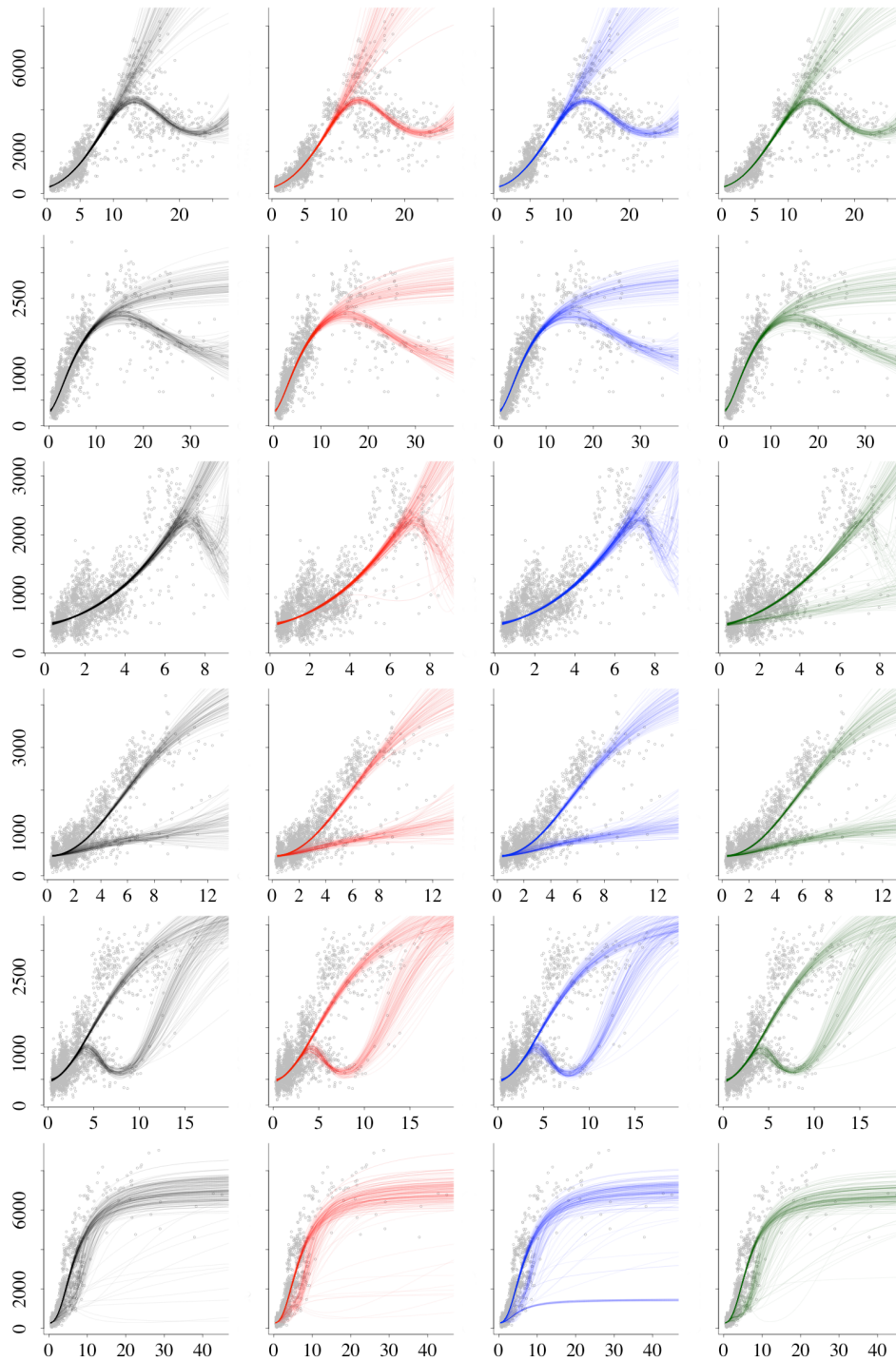
Figure 11: Comparison of the local fits of the full MCMC computation (black) for the astronomy example and the final distributed EP approximations when the groups are distributed into $K = 5$ (red), $K = 10$ (blue), and $K = 30$ (green) sites. Posterior draws are shown for each of 6 groups (one group per row) with longitudes $12°, 32°, 82°, 92°, 93°$, and $194°$.
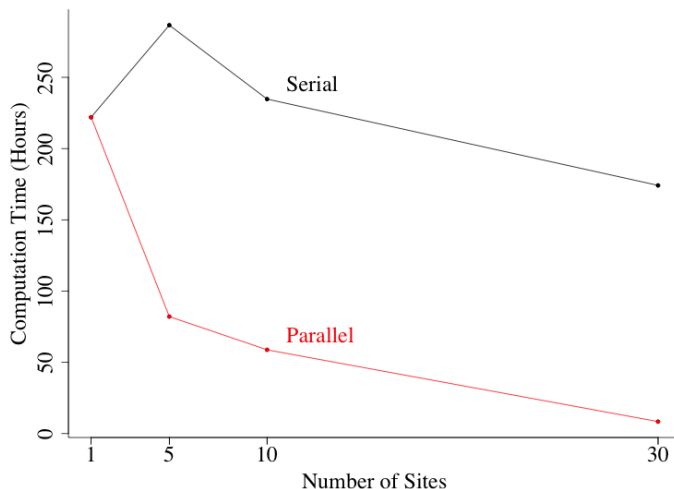
Figure 12: Computation times for the distributed EP algorithm applied to the astronomy data, as a function of the number of sites. The full MCMC computation time is equivalent to that of EP with $K = 1$ site. The computational benefits of increasing the number of sites is clear when the updates are parallel.

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°.

Figure 12 illustrates the computation times for the EP runs with serial and parallel updates. The advantages of distributed EP are most clear when comparing K = 1 site to K = 30 sites, which 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.

7. Discussion

Using the principle of message passing with cavity and tilted distributions, we have 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. Furthermore, EP comes with the additional advantage of naturally sharing information between distributed parts, focusing the computation into important areas of the parameter space. In our experiment, the method outperforms comparable consensus MC algorithm (Scott et al., 2016) both in time and approximation error.

Probabilistic programming languages such as Stan (Stan Development Team, 2017) provide convenient generalizable tools for statistical data analysis. When dealing with problems where the data does not fit in the memory, the EP framework can be included in

the process to distribute the computation without loosing the generalizability. >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.

The message passing framework presented in this paper includes numerous design choices, and many methods can be subsumed under it. In particular, the inference in the sites can be implemented in various ways. This extensive configurability provides possibilities for improved efficiency but also makes it more complex to set up. In this paper we have discussed two generalizable simulation based methods in particular, moment matching and SNEP (Hasenclever et al., 2017). These methods perform better in different situations. It is also possible to use them in combination, for example using moment matching in early iterations for a more stable and quicker start, and SNEP in later iterations for more stable and precise convergence. If convergence problems are encountered, while different choices in the method may also be helpful, it could be useful to first consider alternative approximating families instead. Further 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., 2017; Hasenclever et al., 2017; Bardenet et al., 2017). 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 in a way that achieves the computational benefits of parallelism while allowing each local update to make use of relevant information from the other sites.

While EP may not yet be a "way of life," we argue that the increasing popularity of divide-and-conquer algorithms in big data environments is moving us in this direction. Stepping back from particular choices in implementation, the idea of the cavity and tilted distributions seems to us to be crucial in understanding how inferences from separate pieces of information can be combined in a way that respects the model being fit. We anticipate that great progress could be made by using message passing to regularize existing algorithms.

Acknowledgments

Appendices

A. Distributed Parallel Message Passing Algorithm

This section presents a detailed algorithm for distributed EP applied in the context of partitioned data. The implementation used in the experiments in Section 6 follows this description. In Appendix A.5, we further extend the algorithm for cases where dimension reduction is possible.

A.1. Algorithm Description

In this subsection we give a practical algorithm description suitable for implementing the general message passing algorithms discussed in Sections 3 and 4. The algorithm can be applied to approximate the joint posterior distribution in a general setting or the marginal posterior distribution of the shared parameters in a hierarchical setting.

Consider the normal distribution $g(\theta|r,Q) = \mathrm{N}(\theta|\mu,\Sigma)$ for the random variable $\theta \in \mathbb{R}^D$. The precision mean vector $r \in \mathbb{R}^D$ and the symmetric (positive semidefinite) precision matrix $Q \in \mathbb{R}^{D \times D}$ are the natural parameters and the mean vector $\mu \in \mathbb{R}^D$ and the symmetric (positive semidefinite) covariance matrix $\Sigma \in \mathbb{R}^{D \times D}$ are the moment parameters. The parameters can be inverted from natural to moment form, $\Sigma = Q^{-1}$, $\mu = Q^{-1}r$, and vice-versa, $Q = \Sigma^{-1}$, $r = \Sigma^{-1}\mu$, using Cholesky factorization and backward substitution. Multiplying together two normal distributions yields an unnormalized normal distribution with natural parameters multiplied together $g_1(\theta|r_1,Q_1)g_2(\theta|r_2,Q_2) \propto g_{1\cdot2}(\theta|r_1+r_2,Q_1+Q_2)$, and analogically $g_1(\theta|r_1,Q_1)/g_2(\theta|r_2,Q_2) \propto g_{1/2}(\theta|r_1-r_2,Q_1-Q_2)$ (e.g., Rasmussen and Williams, 2006, p. 200).

EP is applied to approximate the target posterior distribution,

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

by a normal distribution,

$$g(\theta|r,Q) \propto g_0(\theta|r_0,Q_0) \prod_{k=1}^{K} g_k(\theta|r_k,Q_k),$$

where the global approximation $g(\theta|r,Q)$, site approximations $g_k(\theta|r_k,Q_k)$, and the prior $g_0(\theta|r_0,Q_0)$ are all normal distributions parameterized by mean vector and precision matrix parameters. The global approximation parameters r,Q can be obtained by summing up all the site parameters and the prior parameters:

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

In the following algorithm description, the parameter $\eta \in (0,1]$ can be used to apply power EP (Minka, 2004) to minimize general α-divergence instead of KL divergence as discussed in Section 5.1 in the paper. Using $\eta = 1$, as we did in our experiments, applies regular EP with KL divergence minimization.

Initially all the site distributions are set to improper uniform distributions with $r_k = 0$, $Q_k = 0$ for $k = 1, 2, \ldots, K$, which is equivalent to initializing the global approximation $g(\theta|r, Q)$ to the prior, that is, $r = r_0$ and $Q = Q_0$. The algorithm proceeds by iteratively updating the site distributions until convergence:

1. In parallel at each site $k = 1, 2, \ldots, K$, determine the cavity distribution $g_{-k}(\theta|r_{-k}, Q_{-k})$:

$$Q_{-k} = Q - \eta Q_k, \qquad r_{-k} = r - \eta r_k.$$

   Here it is possible to obtain a precision matrix Q_k that corresponds to an improper distribution (not in the first iteration). This is acceptable as long as the inference for the tilted distribution in the next step can be carried out. If this inference method requires a proper cavity distribution, the algorithm can jump to step 4, reduce damping, and continue until proper cavities are obtained.

2. In parallel at each site $k = 1, 2, \ldots, K$, approximate the natural precision parameters $r_{\backslash k}, Q_{\backslash k}$ of the tilted distribution

$$g_{\backslash k}(\theta) \propto p(y_k|\theta)^\eta g_{-k}(\theta|r_{-k}, Q_{-k}),$$

   which is of unrestricted form. This can be sampled and differentiated using

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

   In a hierarchical setting, as discussed in Section 4, the tilted distribution considers also the local parameters:

$$g_{\backslash k}(\theta) \propto \int \left(p(y_k|\alpha_k, \theta)p(\alpha_k|\theta)\right)^\eta g_{-k}(\theta|r_{-k}, Q_{-k}) \, \mathrm{d}\alpha_k,$$

   where θ contains the shared parameters and α_k contains the local parameters for site k.

   Key properties of different approximation methods are:

   - MCMC: It is easy to compute $\mu_{\backslash k}$ and $\Sigma_{\backslash k}$ from a set of simulation draws. Various approaches for computing the precision matrix $Q_{\backslash k} = \Sigma_{\backslash k}^{-1}$ are discussed in Appendix A.3 and in Section A.4.

   - 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

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

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

Other approximation methods can also be used, including EP itself, which can be used to form arbitrarily deep and complex message passing algorithms.

The implementation used in the experiments in Section 6 uses MCMC sampling and consider the hierarchical structure of the problem.

3. In parallel at each site $k = 1, 2, \ldots, K$, if $|Q_{\backslash k}| > 0$, compute the change in the site distribution $g_k(\theta | r_k, Q_k)$ 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 some method discussed in Section 5.3 to improve the conditioning of $Q_{\backslash k}$ and compute the parameter updates with the modified $Q_{\backslash k}$.

4. Update the global approximation $g(\theta | r, Q)$ with damping level $\delta \in (0, 1]$:

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

If the resulting approximation $g(\theta | r, Q)$ is not proper, decrease δ and try again.

5. In parallel at each site $k = 1, 2, \ldots, K$, determine the updated site parameters with the selected damping level $\delta \in (0, 1]$:

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

The iterations are repeated until all the tilted distributions are consistent with the approximate posterior, that is, Δr_k and ΔQ_k become small for all sites $k = 1, 2, \ldots, K$.

A.2. Advantages and Limitations

This section discusses advantages and limitations of the algorithm presented in Appendix A.1.

A.2.1. Advantages

- Working with the natural parameters of the exponential family makes the computations in the algorithm convenient. Operating with such terms can be parallelized elementwise, making the time complexity constant instead of $O(D^2)$, for example $[Q_1 + Q_2]_{i,j} = [Q_1]_{i,j} + [Q_2]_{i,j}$, where $[A]_{i,j}$ denotes element i, j of matrix A. Also, summing up multiple terms in step 4 can be parallelized termwise, for example $(Q_1 + Q_2) + (Q_3 + Q_4)$.

- The tilted moments can be determined by sampling directly from the unnormalized tilted distributions or by using 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).

A.2.2. 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 A.4. 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 Appendix A.3, the QR-decomposition can be used here to more efficiently form the Cholesky factor of the matrix directly from the sample.

- Estimating the marginal likelihood is more challenging, because determining the normalization constants of the tilted distribution requires multivariate integrations. For example, annealed importance sampling type of approaches could be used if marginal likelihood estimates are required. We further discuss marginal likelihood in Appendix E.

  With Laplace's method, approximating the this normalization constant is straightforward but the quality of the marginal likelihood approximation is not likely to work well 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.3. 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 scatter matrix (unnormalized sample covariance 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 sample without ever forming the scatter matrix itself. This makes the process more stable, as forming the scatter matrix squares the condition number.

Consider the sample concatenated as an $n \times d$ matrix D where the columns are centered to have zero mean. The scatter matrix is $S = D^T D$. 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.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. In the message passing algorithm, these parameters are needed in natural form. Estimating the moment parameters from a given sample is straightforward but estimating the precision matrix from a set of simulation draws is a complex task and in general the estimators are biased. In addition, depending on the situation, using the naive unbiased moment estimator can produce higher expected KL divergence compared to other biased estimators.

In the algorithm, the parameters of the tilted distribution are needed in natural form. 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 posterior simulation draws representing the tilted distribution. This estimator is biased in general: $\mathrm{E}(\widehat{Q}) \neq Q$. Furthermore, the number of draws n affects 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 could 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.3.

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

$$\widehat{Q}_{\mathrm{N}} = \frac{n-d-2}{n-1}\widehat{\Sigma}^{-1} = (n-d-2)S^{-1}. \tag{5}$$

Furthermore, the precision mean is given by,

$$\widehat{r}_{\mathrm{N}} = \widehat{Q}_{\mathrm{N}}\widehat{\mu} = (n-d-2)S^{-1}\widehat{\mu}, \tag{6}$$

which can be solved simultaneously while inverting the scatter matrix. The inverse of this matrix is a biased estimate of the covariance matrix. Xu et al. (2014) used this estimator in their implementation of the MCMC based EP algorithm and conjectured that it would be an unbiased estimate of the true natural parameters of the tilted distribution, which is not true in general. 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). However, if the tilted distribution is normally distributed, it is likely that the moments can be solved analytically and sample based estimates are not needed in the first place. 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).

In each iteration of EP, as discussed in Section 2, the objective is to minimize the KL divergence from the global approximation to the tilted distribution, which corresponds to matching the moments. With approximated moment estimates, in order to make the algorithm work like EP on expectation and stay at the same fixed point as EP on expectation, the moment estimates should be unbiased. However, the distribution of the resulting KL
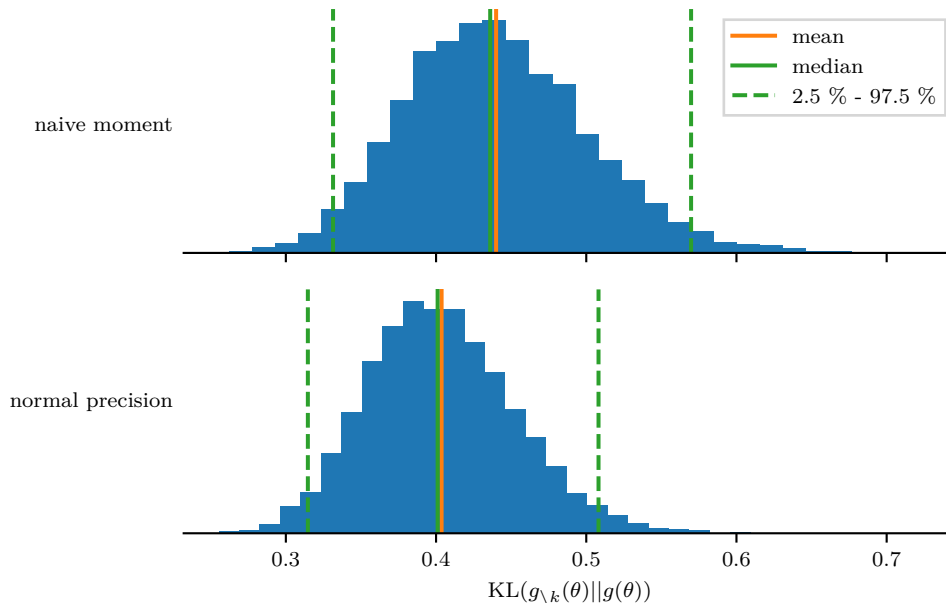
Figure 13: Simulated histograms of resulting KL divergences $\mathrm{KL}(g_{\backslash k}(\theta)||g(\theta))$ when moment parameters are estimated using the naive unbiased moment estimator $S/(n-1)$ and the normal natural estimator $S/(n-d-2)$ presented in Equations (5) and (6). The sample size for estimating the parameters $n = 200$. The tilted distribution is normally distributed with $d = 16$ dimensions and the correlation matrix is randomized with eigenvalues drawn from a Dirichlet distribution with a few higher values in the concentration parameter. The sampling is repeated 8000 times to form a sample of the distribution of the KL divergences. It can be seen from the histograms and from the illustrated statistics, that the biased estimator performs better in this case.

divergence depends on the used moment parameter estimator. An unbiased estimator is not necessarily optimal, as there can be biased estimators that produce smaller expected KL divergence. The situation depends on the form of the tilted distribution and on the properties of the used estimators, for example the sample size and consistency in the case of simulation-based estimates. Figures 13 and 14 illustrates a simulated example case, where the tilted distribution moments are estimated from a sample using various scatter matrix based estimators. Figure 13 illustrates the distribution of the resulting KL divergence $\mathrm{KL}(g_{\backslash k}(\theta)||g(\theta))$ with normal tilted distribution $g_{\backslash k}(\theta)$ using the naive unbiased estimate and the normal distribution natural parameter estimates presented in Equations (5) and (6). In this case, the biased moment estimator outperforms the unbiased naive one. In Figure 14, a t distribution with four degrees of freedom is used. In this case, the optimal scaling of the scatter matrix is greater than in both of the discussed estimators. Thus, when implementing a simulation-based EP algorithm, it may be beneficial to study the form of
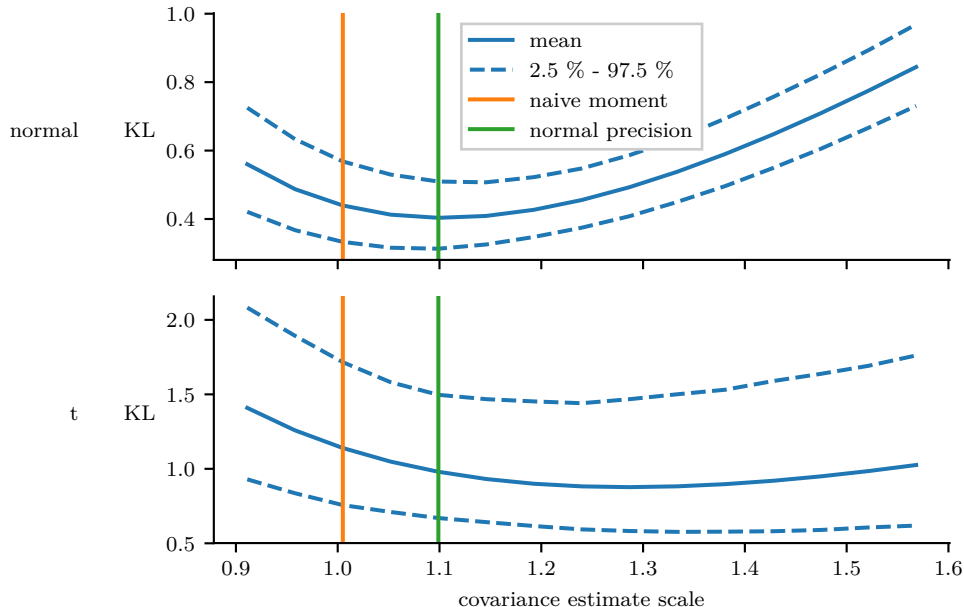
Figure 14: Simulated statistics of resulting KL divergences $\mathrm{KL}(g_{\mathrm{EP}}(\theta)||g_{\mathrm{sample}}(\theta))$, where $g_{\mathrm{EP}}(\theta)$ is the new global distribution with precise EP update and $g_{\mathrm{sample}}(\theta)$ is the new global distribution with sample based update. In the sample based update, different scaling of the scatter matrix is used, indicated in the x-axis as multiples of $1/n$. The naive unbiased moment estimator $S/(n-1)$ and the normal natural estimator $S/(n-d-2)$ presented in Equations (5) and (6) are indicated by vertical lines. The sample size for estimating the parameters $n = 200$. In the first row, the $d = 16$ dimensional tilted distribution is normally distributed, and in the second row, it follows t-distribution with four degrees of freedom. The correlation matrix is randomized in a similar fashion as in Figure 13. The sampling is repeated 2000 times to form a sample of the distribution of the KL divergences for each estimator in a grid of 15 points. It can be seen that while producing best results in the normal case, the normal natural estimator does not perform best in the t-distribution case.

the tilted distribution and select the used sample estimator accordingly. The need for this selective analysis can be hindered by increasing the sample size; With increased sample size, the naive moment and the normal natural estimators are likely to perform equally well.

A.5. Dimension Reduction for Site Inference

The EP algorithm presented in Appendix A.1 can easily be extended to incorporate additional message passing components, as discussed for example by Chen and Wand (2020). Here we demonstrate a version for the special case in which the non-Gaussian likelihood terms $p(y_k|\theta)$ depend on θ only through preferably low-dimensional linearly transformed random

variables $z_k = U_k^T \theta$, $U_k \in \mathbb{R}^{D \times D_k}$, $z_k \in \mathbb{R}^{D_k}$ for each partition $k = 1, 2, \ldots, K$; that is, $p(y_k|\theta) = p(y_k|z_k)$.

In the algorithm, the site approximations are stored in low-dimensional form (zero initialized) as $g_k(z_k|\tilde{r}_k, \widetilde{Q}_k)$, $\tilde{r}_k \in \mathbb{R}^{D_k}$, $\widetilde{Q}_k \in \mathbb{R}^{D_k \times D_k}$. The global approximation $g(\theta|r, Q)$ in the original space can be obtained from the transformed site distributions and the prior distribution (e.g., Chen and Wand, 2020, section Multivariate Linear Combination Derived Variable Fragment)

$$Q = Q_0 + \sum_{k=1}^{K} Q_k \quad r = r_0 + \sum_{k=1}^{K} r_k,$$

where

$$Q_k = U_k \widetilde{Q}_k U_k^T, \qquad r_k = U_k \tilde{r}_k. \tag{7}$$

The algorithm proceeds similarly as in Appendix A.1, but the site updates are performed in the transformed space:

- In step 1, the cavity distribution $g_{-k}(z_k|\tilde{r}_{-k}, \widetilde{Q}_{-k})$ is calculated by,

$$\widetilde{Q}_{-k} = \left(U_k^T Q^{-1} U_k \right)^{-1} - \eta \widetilde{Q}_k, \qquad \tilde{r}_{-k} = \left(U_k^T Q^{-1} U_k \right)^{-1} U_k^T Q^{-1} r - \eta \tilde{r}_k,$$

  see for example Chen and Wand (2020, Section Multivariate Linear Combination Derived Variable Fragment).

- In steps 2, 3, and 5, the computations are applied for z_k, $\tilde{r}_k$, and $\widetilde{Q}_k$ instead of θ, r_k, and Q_k.

- In step 4, the global approximation is updated by transforming the difference into the original space by considering the relation in (7):

$$Q^{\text{new}} = Q + \delta \sum_{k=1}^{K} \Delta Q_k = Q + \delta U_k \left[\sum_{k=1}^{K} \Delta \widetilde{Q}_k \right] U_k^T$$

$$r^{\text{new}} = r + \delta \sum_{k=1}^{K} \Delta r_k = r + \delta U_k \left[\sum_{k=1}^{K} \Delta \tilde{r}_k \right].$$

The advantage of the algorithm comes from the lower dimensional operation of the site updates; the tilted distribution inference considers only the transformed space of z_k with D_k dimensions instead of the space of θ with D dimensions. In addition, the site distributions can be stored in the lower dimensional space with $O(D_k^2)$ elements in parameters $\tilde{r}_k, \widetilde{Q}_k$ instead of the original space with $O(D^2)$ elements in parameters r_k, Q_k.

The disadvantage of the method is that the computation of the cavity distributions becomes heavier task with time complexity $O(D^3)$. However, if the global approximation moment parameters $\mu = Q^{-1}r$ $\Sigma = Q^{-1}$ are solved between every iteration anyway, for example to monitor the convergence, the task does not add any complexity.

B. Additional Details of the Experiments

As discussed in the text, for example in Section 5, implementing a distributed EP algorithm contains multiple design choices that might affect the behaviour of the algorithm. This section describes some details of the implementation we used in our experiments in Section 6. In addition, we show some additional results obtained from the simulated experiment. The algorithm follows the description in Appendix A.1, where the tilted distribution inference in step 2 considers the discussed hierarchical setting and is carried out by MCMC sampling.

B.1. Implementation in Stan

We implement our experiments using Python, R, and Stan. The Python code for the simulated experiment is available at https://github.com/gelman/ep-stan/releases/tag/v1.3. We pass the normal approximations g_k back and forth between a master node and K separate site nodes. In the site nodes, we use Stan to compute the tilted distribution moments.

Our implementations are not optimal in methodological point-of-view. In the following, we list some key areas of improvement for our implementation:

- In the current implementation, we write the appropriate Stan model for the tilted distribution inference manually by adapting the code from the full model to act just on the subset of parameters relevant to a single subset of the partitioned hierarchical model. In future software development, we would like to be able to take an existing Stan program and merely overlay a factorization so that the message passing algorithm could be applied directly.

- Currently, Stan performs adaptation each time it runs. Future versions should allow restarting from the previous state, which should speed up computation substantially when the algorithm starts to converge.

- We should be able to approximate the expectations more efficiently using importance sampling.

B.2. Simulated Experiment Marginal KL Divergence

In the simulated hierarchical logistic regression experiment in Section 6.1, the bottom subplot of Figure 6 shows the KL divergence from the reference posterior distribution to the approximate one. In addition to the mean and variance of the parameters, this measure also takes into account the correlations between the parameters. Figure 15 features an analogous plot, where the correlation between the parameters is ignored; instead of the full KL divergence, the sum of the KL divergences between all the marginals is illustrated. With this measure, the difference in the accuracy between the EP approximation and the consensus MC approximation becomes bigger.

B.3. Details of the Simulated Hierarchical Logistic Regression

In the simulated hierarchical logistic regression problem in Section 6.1, the difficulty of the problem is controlled by regulating the resulting uncertainty in the simulated data. This is done by first fixing random model parameter values and then selecting suitable parameters
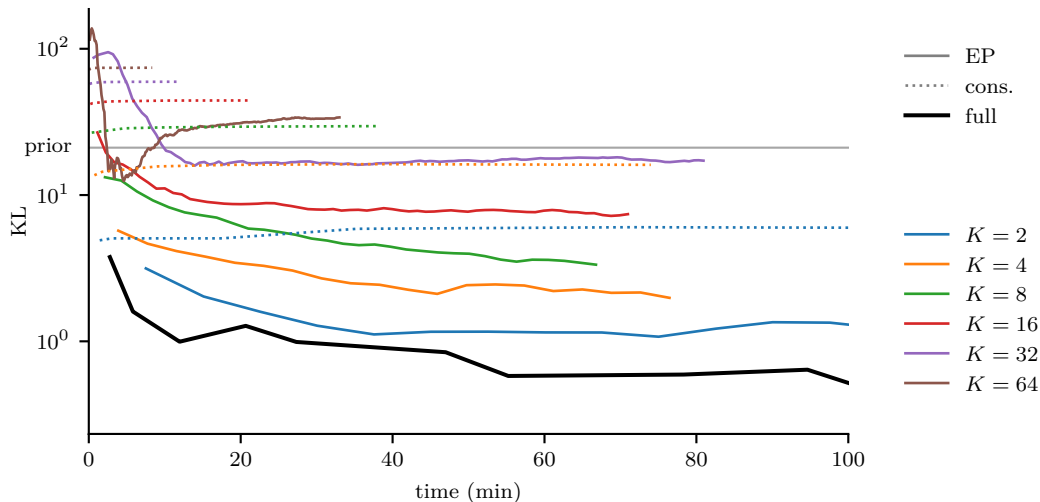
Figure 15: Approximate marginal KL divergence of the posterior approximation from the target distribution as a function of the elapsed sampling time in the simulated hierarchical model. Three methods are compared: full MCMC, distributed EP, and distributed consensus MC. For EP (solid lines) and consensus MC (dotted lines), line colors indicate the number of partitions K. The y-axis is in the logarithmic scale. Compared to the full KL divergence illustrated in the analogous plot in Figure 6, the difference in the accuracy between EP and consensus MC is bigger.

for sampling the explanatory variable conditional to the model parameters for each group separately. Finally the response variable y_{ij} is sampled given the explanatory variable x_{ij} and group parameter β_j. The following describes these steps in detail.

First the hyperparameters are fixed. For the intercept, we set $\mu_0 = 1.5$ and $\log \sigma_0 = 0.4$, and for the slope, μ_d and $\log \sigma_d, d = 1, 2, \ldots, D$ are drawn at random from uniform$(-2, 2)$ and uniform$(-0.5, 0.5)$ respectively. The prior for the hyperparameters is set so that $\mu_d \sim \mathrm{N}(0, 4^2)$ and $\mu_d \sim \mathrm{log\text{-}N}(0, 2^2)$ for $d = 0, 1, \ldots, D$. Given the fixed hyperparameters, the group parameters β_j are drawn at random according to the model distribution $\beta_{jd} \sim \mathrm{N}(\mu_d, \sigma_d^2)$.

The following describes the sampling of the data based on the fixed parameters β_j. Vectors (lowercase) and matrices (uppercase) are denoted with bold symbols to distinguish them from scalars. In addition, denoting the conditioning on the model parameters $\boldsymbol{\beta}_j$ and the group indexing j is omitted. Detailed derivations of the formulas are presented in Section 6.3.2 in (Sivula, 2015).

The explanatory variable $\boldsymbol{x} \in \mathbb{R}^D$ is sampled from normal distribution $\mathrm{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{\Sigma}_0)$. The mean $\boldsymbol{\mu}$ is restricted to be equal in all dimensions: $\boldsymbol{\mu} = \mu \mathbb{1}$, where $\mathbb{1}$ is a vector of 1's. The correlation structure $\boldsymbol{\Sigma}_0$ is randomly generated using modified vines method by Lewandowski et al. (2009), where the partial correlations are sampled from Beta$(2, 2)$ from the range $(-0.8, 0.8)$ and the diagonal is normalized to unity.

Consider the regression coefficient vector split into intercept coefficient α and slope $\boldsymbol{\beta}$. The classification uncertainty is controlled by setting restrictions to $P = \text{logit}^{-1}(\alpha + \boldsymbol{\beta}^T \boldsymbol{x})$, that is to the probability of the response variable y being in one class, which follows logit-normal distribution $\text{logit-N}(\alpha + \boldsymbol{\beta}^T \boldsymbol{\mu}, \sigma^2 \boldsymbol{\beta}^T \boldsymbol{\Sigma}_0 \boldsymbol{\beta})$. The resulting distribution of P is restricted to have tail probabilities $\Pr(P \leq p_0) < \gamma_0$ and $\Pr(P > p_0) < \gamma_0$, where $p_0 = 0.2$ and $\gamma_0 = 0.01$. In addition, smallest acceptable variance condition $\text{Var}(\alpha + \boldsymbol{\beta}^T \boldsymbol{x}) \geq \tau_{\min}^2$ is set with $\tau_{\min} = 0.25$. Satisfying explanatory variable sampling parameters μ and σ are then chosen by

$$
\mu = \begin{cases}
\dfrac{\delta_{\max} - \alpha}{\sum_{d=1}^{D} \beta_i}, & \text{if } \alpha > \delta_{\max}, \\[2ex]
\dfrac{-\delta_{\max} - \alpha}{\sum_{d=1}^{D} \beta_i}, & \text{if } \alpha < -\delta_{\max}, \\[2ex]
0 & \text{otherwise.}
\end{cases}
$$

$$
\sigma = \begin{cases}
\dfrac{\text{logit}(p_0) + |\alpha|}{\Phi^{-1}(\gamma_0)\sqrt{\boldsymbol{\beta}^T \boldsymbol{\Sigma}_0 \boldsymbol{\beta}}}, & \text{if } |\alpha| \leq \delta_{\max}, \\[2ex]
\dfrac{\tau_{\min}}{\sqrt{\boldsymbol{\beta}^T \boldsymbol{\Sigma}_0 \boldsymbol{\beta}}}, & \text{otherwise,}
\end{cases}
$$

where $\delta_{\max} = \tau_{\min} \Phi^{-1}(\gamma_0) - \text{logit}(p_0)$ is the maximum magnitude of the mean of $\alpha + \boldsymbol{\beta}^T \boldsymbol{x}$. With this parameter selection method,

$$
P \sim \text{logit-N}\left(\alpha', \frac{\text{logit}(p_0) + |\alpha'|}{\Phi^{-1}(\gamma_0)}\right),
$$

where

$$
\alpha' = \min\big(\max(\alpha, -\delta_{\max}), \delta_{\max}\big).
$$

From this, it can be seen that the resulting distribution of P does not depend on $\boldsymbol{\beta}$ or the dimensionality D. The parameter α tilts the distribution toward 0 or 1.

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 with a multivariate normal approximating family, where the likelihood is factored into K sites. The tilted distribution is approximated with MCMC sampling. Let N_k be the number of data points in site k and let D be the number of parameters, that is, the length of the vector θ. 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 sites. The central unit stores the global approximation and the worker units

store the respective site approximation. Each distribution parameters consist of $O(D^2)$ values: mean or precision mean vector of length D and covariance or precision matrix of size $D \times D$. Furthermore, we assume a network transmission cost of c per parameter. Finally, we define $h(n, d)$ as the computational cost of generating a sample from a tilted distribution with n data points and d parameters. In general case $h(n, d) \gg O(d^2 + n)$, where $O(d^2 + n)$ would be the minimal cost for analytically tractable case.

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 initialization of the approximations can be performed in parallel in every unit. Here it is assumed, that single parameter allocation is a constant time operation.

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. First, the current global approximation needs to be sent from the master node to the worker nodes with cost $O(cKD^2)$. On the worker nodes in parallel, this step involves only simple subtraction of $O(D^2)$ values per site, which can be parallelized locally. Thus the resulting total cost is $O(cKD^2)$.

   (b) Fitting an updated local approximation $g_k(\theta)$. This step is performed on parallel in every worker. First a sample from the tilted distribution is generated with cost $h(N_k, D)$. After this, moment estimates are generated in natural form based on the obtained sample. As discussed in Appendix A.3, QR-decomposition is used to form the estimates with cost $O(D^3)$. The resulting total cost is $O(h(\max N_k, D) + D^3)$.

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

   (d) Update the global approximation $g(\theta)$. Summing up the updates from all the sites has the naive cost of $O(KD^2)$, or improved cost of $O(\log K)$, when summations are parallelized elementwise and termwise. However, if the cost h of approximating the posterior distribution is variable across worker units, the central unit could update $g(\theta)$ whenever possible while waiting for other sites to finish.

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:

$$O\Big(m\Big(cKD^2 + h(\max N_k, D) + D^3\Big)\Big).$$

By comparison, consider first the cost of a non-parallel version:

$$O\Big(m\Big(KD^3 + \textstyle\sum_k h(N_k, D)\Big)\Big).$$

Second, consider the cost of full sampling with no partitioning:

$$O\Big(h\Big(\sum_k N_k, D\Big)\Big).$$

With these three expressions, we can immediately see the computational benefits of our scheme. In many cases, sampling will be by far the most costly operation, and will depend superlinearly on its arguments. Thus, the parallel message passing scheme will dominate. As the total data size $N = \sum_k N_k$ grows large, our scheme becomes essential. When data is particularly big (for example $N \approx 10^9$), our scheme will dominate even in the rare case that $h(n, d)$ is in its minimal $O(d^2 + n)$.

D. Comparison of SNEP and Moment Matching

Section 5.1 discussed the SNEP method introduced by Hasenclever et al. (2017), which can be used for a simulation-based site inference instead of the tilted distribution moment matching method. In this section, we shortly compare both methods in two experiments. The experiments do not give an exhaustive view on the difference of the behaviour of the methods in all situations. However, they show that both of the methods have pros and cons and that the preferability of the methods is situational. The source code for both experiments is available online at https://github.com/gelman/ep-stan.

We replicate the simulated hierarchical logistic regression experiment on Section 6.1 in a smaller scale with $J = 16$ hierarchical groups and $D = 3$ dimensions of the explanatory variable. The problem is distributed into $K = 2, 4, 8, 16$ sites. We apply a constant damping factor of 0.8 for SNEP and 0.5 for moment matching. We apply four inner iterations with auxiliary parameter updating in every other iteration in the site update for SNEP. Because SNEP is not compatible with uniform initial site distributions, these distributions were set to $N(0, 2K \max(\text{Cov}(p(\theta)))I)$ instead. All other settings were identical to the bigger experiment of Section 6.1. The results of the simulation are illustrated in Figure 16, which shows the accuracy of the approximation as a function of the elapsed sampling time. As with Figure 6 for the bigger experiment in Section 6.1, in order to focus on the significant portion of the algorithm and ignore some implementation dependent factors, we do not in this graph count time spent in non-sampling parts of the algorithm. Compared to the moment matching method, the site update portion of SNEP contains more heavy operations, such as Cholesky factorizations of order d_ϕ, and consequently SNEP spent more time in non-sampling parts of the algorithm in our experiment. When the problem was distributed to $K = 2$ sites, SNEP algorithm failed in all sites after two iterations by resulting in an improper global approximation and the iteration had to be terminated. In other runs, both methods converge to a relatively similar solution but not into identical one. Moment matching converged faster than SNEP but had more variability in the end. SNEP was slower and had more variability in the start.

In addition to the single-run simulations, we also set up an experiment for testing the effect of single site updating in different settings. We set up a randomized three-dimensional normal target distribution consisting of six normal factorized parts with variances ranging from e to e^{-1} and a normal prior distribution $p(\theta) \sim N(0, 4^2 eI)$. Each factor is approximated in one site. We measure the resulting global approximation KL divergence after performing
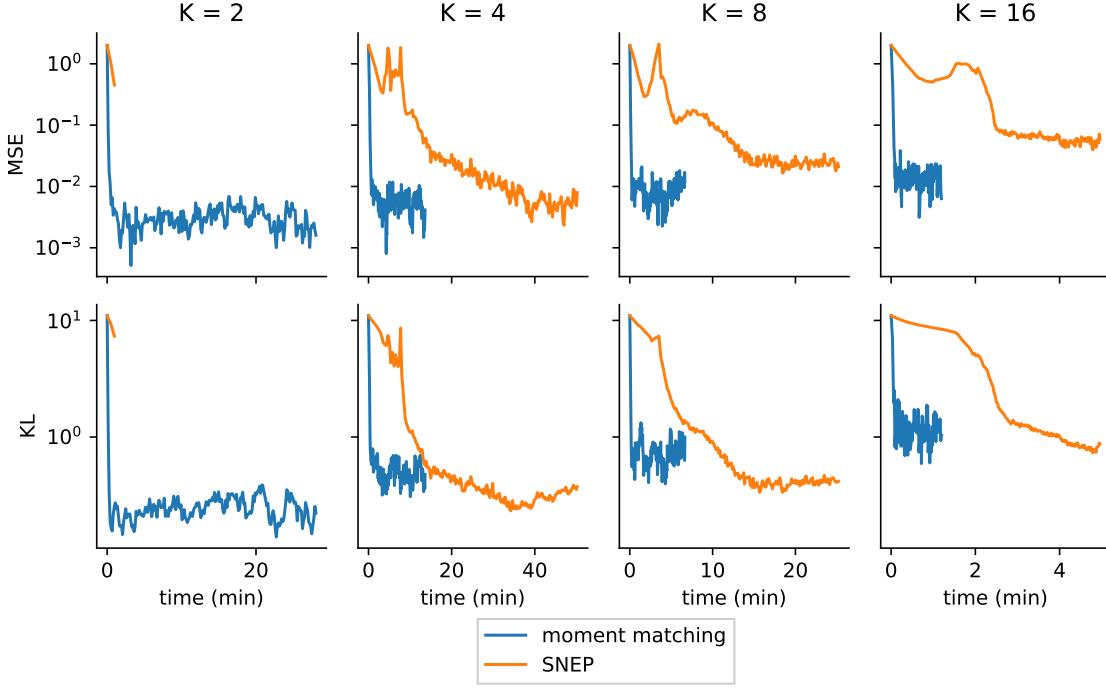
44

Figure 16: Comparison of simulation-based moment matching EP and SNEP in a simple hierarchical logistic regression experiment. The plots show the mean squared error of the mean and approximate Kullback-Leibler divergence from the target distribution to the resulting posterior approximation as a function of the elapsed sampling time. The y-axis is in the logarithmic scale. Each column corresponds to a different number of partitions K. When $K = 2$, SNEP results in an improper global approximation and fails after two iterations. In other runs, both methods converge to a solution with comparable accuracy. SNEP reaches the solution slower and has more variability in the beginning. Moment matching has a faster start but more variability in the end.

one site update with different initial site and cavity distributions. These distributions are set

$$g_k(\theta) = \left(f_k(\theta)\right)^{1-d_{\text{site}}} \left(g_k^{\text{extreme}}(\theta)\right)^{d_{\text{site}}}$$

$$g_{-k}(\theta) = \left(\prod_{i \neq k} f_i(\theta)\right)^{1-d_{\text{cavity}}} \left(g_{-k}^{\text{extreme}}(\theta)\right)^{d_{\text{cavity}}},$$

where d_{site} and d_{cavity} control the deviation from the converged state, $f_k(\theta)$ is the true underlying factor and the site distribution at the convergence, and $\prod_{i \neq k} f_i(\theta)$ is the cavity distribution at the convergence. The extreme site and cavity distributions are set to $g_k^{\text{extreme}}(\theta) \sim \text{N}(0, 2^2 \text{eI})$ and $g_{-k}^{\text{extreme}}(\theta) = \left(g_k^{\text{extreme}}(\theta)\right)^{K-1} p(\theta)$ respectively. For both methods, the sample size at the site update is 200. For SNEP, we apply one inner iteration
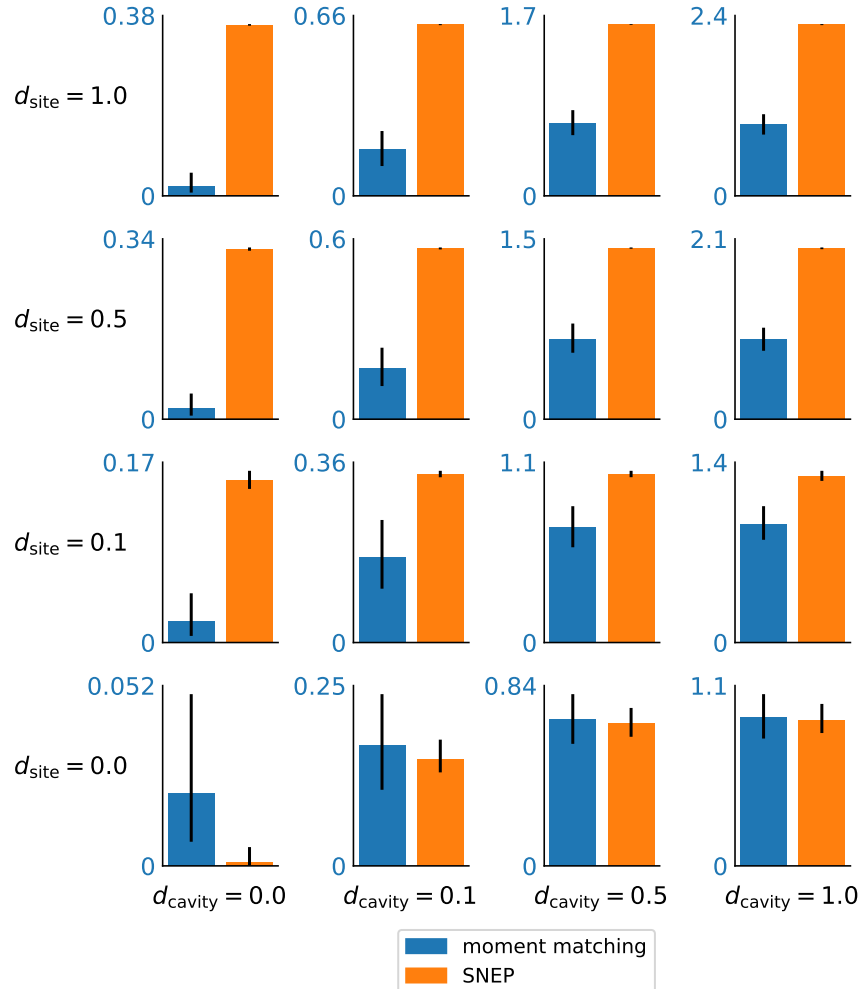
Figure 17: KL divergence from the target distribution to the resulting posterior approximation after updating one site with different initial site settings. In each row and column, d_{site} and d_{cavity} indicate the divergence of the respective initial distribution from the convergence, where 0.0 corresponds to the converged state and 1.0 to a diverged state. In order to focus on the difference between the methods, the y-axis is in different scale in each tile. The simulation is repeated 2000 times. The bar heights show the medians of the obtained resulting KL divergences, and the attached black lines indicate 2.5% and 97.5% quantiles. It can be seen from the figure that moment matching method has greater variability in general. In this case, moment matching outperforms SNEP when the site distribution is not completely converged.

for the site update. We replicate the experiment 2000 times. The results of the experiment are illustrated in Figure 17. Experimenting with four inner SNEP iterations with auxiliary parameter update in every other iteration did not change the results much.

Based on the discussed two experiments, it is clear that the methods behave differently and have characteristics that make them better in different situations. The experiments suggest that SNEP behaves more chaotically and progresses slower when far away from the convergence. On the other hand, it seems to be more stable when sufficiently close to the convergence. We believe this behavior is related to SNEP operating on the moment domain instead of the natural domain as it is done in moment matching. To the best of our knowledge, this feature makes SNEP also incompatible with uniform initial site distributions. One interesting idea would be to apply moment matching in early iterations for possibly more stable and faster start and switch to using SNEP for later iterations for more stable convergence. Further study is needed in order to draw more elaborate conclusions about their differences.

E. 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, $p(y) = \int p_0(\theta)p(y|\theta)\,d\theta$. This quantity is often used in model choice.

To this end, associate a constant Z_k to each approximating site $g_k(\theta)$ 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 $g_k(\theta) = e^{-\frac{1}{2}\theta^T Q_k \theta + r_k^T \theta}$, under natural parameterization, and denote by $\Psi(r_k, Q_k)$ the corresponding normalizing constant:

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

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 $g_{\backslash k}(\theta)$; (r, Q) is the natural parameter of $g(\theta)$; and $r = \sum_{k=1}^{K} r_k$, $Q = \sum_{k=1}^{K} Q_k$, $r_{-k} = \sum_{j \neq k} r_j$, $Q_{-k} = \sum_{j \neq k} Q_j$. For deterministic approaches, we have discussed approximating the moments of $g_{\backslash k}(\theta)$, 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 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.

F. Optimizing EP Energy

Consider an EP approximation in the exponential family distribution,

$$g(\theta) = \frac{1}{Z} p(\theta) \prod_{k=1}^{n} g_k(\theta),$$

where Z is a normalization constant. The global approximation can be formulated as $g(\theta) \propto p(\theta) \exp(\boldsymbol{s}^T \boldsymbol{\lambda})$ and the cavity distribution as $g_{\backslash k}(\theta) \propto p(\theta) \exp(\boldsymbol{s}^T \boldsymbol{\lambda}_{\backslash k})$, where $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_{\backslash k}$ denote natural parameters. A fixed point of the EP algorithm corresponds to a stationary point of the following objective function (Minka, 2001b):

$$\min_{\lambda} \max_{\lambda_{\backslash k}} (K-1) \log \int p(\theta) \exp(\boldsymbol{s}^T \boldsymbol{\lambda}) \, \mathrm{d}\theta - \sum_{k=1}^{K} \log \int p(\theta) p(y_k|\theta) \exp(\boldsymbol{s}^T \lambda_{\backslash k}) \, \mathrm{d}\theta$$

$$\text{such that} (K-1)\lambda = \sum_{k=1}^{K} \lambda_{\backslash k}.$$

This objective function corresponds to $-\log Z$ and to the expectation-consistent approximation (Opper and Winther, 2005). The correspondence and connection to the Bethe free energy is demonstrated by Heskes et al. (2005).

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*, pages 1591–1598, 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)*, pages 181–189, 2014.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(47):1–43, 2017.

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. *Contributions in infinite-dimensional statistics and related topics*, pages 55–60, 2014.

Wilson Y. Chen and Matt P. Wand. Factor graph fragmentization of expectation propagation. *Journal of the Korean Statistical Society*, 2020. doi: https://doi.org/10.1007/s42952-019-00033-9.

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.

Botond Cseke, Manfred Opper, and Guido Sanguinetti. Approximate inference in latent Gaussian-Markov models from continuous time observations. In *Advances in Neural Information Processing Systems*, pages 971–979, 2013.

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. Expectation propagation performs a smoothed gradient descent. *arXiv preprint arXiv:1612.05053*, 2016.

Guillaume Dehaene and Simon Barthelmé. Expectation propagation in the large data limit. *Journal of the Royal Statistical Society: Series B*, 80(1):199–217, 2018.

Guillaume P Dehaene and Simon Barthelmé. Bounding errors of expectation-propagation. In *Advances in Neural Information Processing Systems 28*, pages 244–252. 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. *Foundations and Trends in Theoretical Computer Science*, 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, 2014a.

Andrew Gelman, Aki Vehtari, Pasi Jylänki, Christian Robert, Nicolas Chopin, and John P. Cunningham. Expectation propagation as a way of life. *arXiv preprint arXiv:1412.4869v1*, 2014b.

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, 2nd edition, 2013.

Erika T. Hamden, David Schiminovich, and Mark Seibert. The diffuse galactic far-ultraviolet sky. *The Astrophysical Journal*, 779(180):15, 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. *Journal of Machine Learning Research*, 18(106):1–37, 2017.

Daniel Hernández-Lobato and Jose Miguel Hernández-Lobato. Scalable Gaussian process classification via expectation propagation. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 168–176, 2016.

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.

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)*, pages 216–223, 2002.

Tom Heskes, Manfred Opper, Wim Wiegerinck, Ole Winther, and Onno Zoeter. Approximate inference techniques with expectation constraints. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11015, 2005.

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.

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. *Journal of Machine Learning Research*, 18 (14):1–45, 2017.

Daniel Lewandowski, Dorota Kurowicka, and Harry Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of multivariate analysis*, 100(9):1989–2001, 2009.

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, Massachusetts Institute of Technology, 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)*, 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)*, 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)*, 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, Dustin Tran, and David Blei. Hierarchical variational models. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 324–333, 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.

Swupnil Sahai. *Topics in Computational Bayesian Statistics With Applications to Hierarchical Models in Astronomy and Sociology*. PhD thesis, Columbia University Academic Commons, 2018.

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*, 11(2):78–88, 2016.

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.

Tuomas Sivula. Distributed Bayesian inference using expectation propagation. Master's thesis, Aalto University, Espoo, Finland, 2015.

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*, 2017. Version 2.17.0, 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 coefficients. *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, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *arXiv:1507.02646*, 2019.

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.