Expectation propagation as a way of life[*]

Andrew Gelman[†] Aki Vehtari[‡] Pasi Jylänki[§] Christian Robert[¶] Nicolas Chopin[‖] John P. Cunningham[**]

15 Dec 2014

Abstract

We revisit expectation propagation (EP) as a prototype for scalable algorithms that partition big datasets into many parts and analyze each part in parallel to perform inference of shared parameters. The algorithm should be particularly efficient for hierarchical models, for which the EP algorithm works on the shared parameters (hyperparameters) of the model.

The central idea of EP is to work at each step with a "tilted distribution" that combines the likelihood for a part of the data with the "cavity distribution," which is the approximate model for the prior and all other parts of the data. EP iteratively approximates the moments of the tilted distributions and incorporates those approximations into a global posterior approximation. As such, EP can be used to divide the computation for large models into manageable sizes. The computation for each partition can be made parallel with occasional exchanging of information between processes through the global posterior approximation. Moments of multivariate tilted distributions can be approximated in various ways, including, MCMC, Laplace approximations, and importance sampling.

Keywords: Bayesian computation, big data, data partitioning, expectation propagation, hierarchical models, Stan, statistical computing

1. Background: Bayesian computation via data partitioning

Various divide-and-conquer algorithms have been proposed for fitting statistical models to large datasets. The basic idea is to partition the data y into K pieces, $y_1, \dots, y_K$, 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 addition to the appeal of such algorithms for parallel or online computation, there is a statistical intuition that they should work well from the law of large numbers: if the K pieces are a random sample of the data, or are independent realizations from a single probability model, then we can think of the log likelihood as the sum of K independent random variables.

In Bayesian inference, one must also decide what to do with the prior distribution, $p(\theta)$. Perhaps the most direct approach is to multiply each factor of the likelihood by $p(\theta)^{1/K}$ and analyze the K pieces separately as $p(y_k|\theta)p(\theta)^{1/K}$, such that the product of the K pieces is the full posterior density. A disadvantage of this approach is that it can lead to instability in the calculation of the fractional posterior distributions. Consider the many settings in statistics and machine learning where an informative prior is needed for regularization—that is, where the likelihood alone does not allow good estimation of θ, with corresponding computational difficulties (for some examples in our

[*] We thank David Blei and Ole Winther for helpful comments, David Schiminovich, Peizhe Li, and Swupnil Sahai for the astronomy example, and the National Science Foundation, Institute for Education Sciences, and Moore and Sloan Foundations for partial support of this research.

[†] Department of Statistics, Columbia University, New York. Visiting Université Paris Dauphine and ENSAE.

[‡] Aalto University, Espoo, Finland.

[§] Donders Institute for Brain, Cognition, and Behavior, Radboud University Nijmegen, Netherlands.

[¶] Université Paris Dauphine and University of Warwick.

[‖] Université Paris Dauphine and ENSAE.

[**] Department of Statistics, Columbia University, New York.

own research, see Gelman, Bois, and Jiang, 1996, and Gelman, Jakulin, et al., 2008, and also in the likelihood-free context, Barthelmé and Chopin, 2014). In these problems, $p(\theta)^{1/K}$ might not provide enough regularization when K is large. Another approach is to include the full prior distribution in each of the K independent inferences, but then this can leads to computational instability when dividing by $p(\theta)^{K-1}$ at the end.

In any case, a challenge in all these algorithms is how to combine the K inferences. If the model is Gaussian, one can simply calculate the posterior mean as the precision-weighted average of the K separate posterior means, and the posterior precision as the sum of the K separate posterior precisions. Minsker et al. (2014) find a median to work well. More generally, though, we will be computing inferences using Monte Carlo simulation. And there is no general way to combine K batches of simulations to get a combined posterior. Indeed, one of our early attempts at data splitting was a bit disappointing, as the gains in computation time were small compared to all the effort required to combine the inferences (Huang and Gelman, 2005).

Various approaches have been recently proposed to combine inferences from partitioned data (see Ahn, Korattikara, and Welling, 2012, Gershman, Hoffman, and Blei, 2012, Korattikara, Chen, and Welling, 2013, Hoffman et al., 2013, Wang and Blei, 2013, Scott et al., 2013, Wang and Dunson, 2013, Neiswanger et al., 2013, and Wang, 2014). These different approaches rely on different approximations or assumptions, but all of them are based on analyzing each piece of the data alone, not using any information from the other $K-1$ pieces.

The contribution of the present paper is to suggest the general relevance of the expectation propagation (EP) framework, leveraging the idea of a "cavity distribution," which approximates the influence of inference from all other $K-1$ pieces of the data, as a prior in the inference step for each partition. Like EP in general, our approach has the disadvantages of being an approximation and requiring a sequence of iterations which are not always stable. The advantage of our approach is that its regularization should induce faster and more stable calculation for each partition. Our approach can be viewed either as a stochastic implementation of EP or, as we prefer, as an application of EP ideas to Bayesian data partitioning.

We present the basic ideas in section 2, but we anticipate that the greatest gains from this approach will come in hierarchical models, as we discuss in section 3. Section 4 considers options in carrying out the algorithm, and in section 5 we demonstrate on a hierarchical logistic regression example. In section 6 we consider various generalizations and connections to other methods, and an appendix contains some details of implementation.

2. A general framework for EP-like algorithms based on iterative tilted approximations

Expectation propagation (EP) is a fast and parallelizable method of distributional approximation via data partitioning. As generally presented, EP is an iterative approach to approximately minimizing the Kullback-Leibler divergence between the true posterior distribution $p(\theta|y)$, and a distribution $g(\theta)$ from a tractable family, typically the multivariate normal. The goal, as is the goal of all approximate inference, is to construct $g(\theta)$ to approximate well the target, $p(\theta|y) \propto p(\theta)\prod_{k=1}^{K} p(y_k|\theta)$.

As we see it, though, the key idea of EP is not specifically about Kullback-Leibler or the evaluation of expectations, but rather that each step works with a "tilted distribution" that includes the likelihood for the kth partition of the data along with the kth "cavity distribution," which represents the approximate posterior for the other $K-1$ pieces. The basic steps are as follows:

1. **Partitioning.** Split the data y into K pieces, $y_1, \ldots, y_K$, each with likelihood $p(y_k|\theta)$.

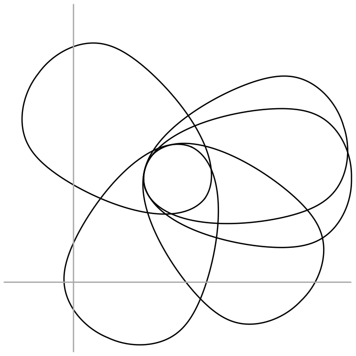

Figure 1: *Sketch illustrating the benefits of expectation propagation (EP) ideas in Bayesian computation. In this simple example, the parameter space θ has two dimensions, and the data have been split into five pieces. Each oval represents a contour of the likelihood $p(y_k|\theta)$ provided by a single 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 $g_{-k}(\theta)$ in a manner inspired by EP, we can devote most of our computational effort to the area of overlap.*

2. **Initialization.** Choose initial site approximations $g_k(\theta)$ from some restricted family (for example, multivariate normal distributions in θ). Let the initial approximation to the posterior density be $g(\theta) = p(\theta)\prod_{k=1}^K g_k(\theta)$.

3. **EP-like iteration.** For $k = 1, \ldots, K$ (in serial or parallel):

   (a) Compute the cavity distribution, $g_{-k}(\theta) = g(\theta)/g_k(\theta)$.

   (b) Form the tilted distribution, $g_{\backslash k}(\theta) = p(y_k|\theta)g_{-k}(\theta)$.

   (c) Construct an updated site approximation $g_k^{\text{new}}(\theta)$ such that $g_k^{\text{new}}(\theta)g_{-k}(\theta)$ approximates $g_{\backslash k}(\theta)$.

   (d) *If parallel*, set $g_k(\theta)$ to $g_k^{\text{new}}(\theta)$, and a new approximate distribution $g(\theta) = p(\theta)\prod_{k=1}^K g_k(\theta)$ will be formed and redistributed after the K site updates. *If serial*, update the global approximation $g(\theta)$ to $g_k^{\text{new}}(\theta)g_{-k}(\theta)$.

4. **Termination.** Repeat step 3 until convergence of the approximate posterior distribution g.

The benefits of this algorithm arise because each site g_k comes from a restricted family with complexity is determined by the number of parameters in the model, not by the sample size; this is less expensive than carrying around the full likelihood, which in general would require computation time proportional to the size of the data. Furthermore, if the parametric approximation is multivariate normal, many of the above steps become analytical, with steps 3a, 3b, and 3d requiring only simple linear algebra. 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, Vanhatalo, and Vehtari, 2011, Cunningham, Hennig, and Lacoste-Julien, 2011, and Vanhatalo et al., 2013), and efforts are made to keep the algorithm stable as well as fast.

Figure 1 illustrates the general idea. Here the data have been divided into five pieces, each of which has a somewhat awkward likelihood function. The most direct parallel partitioning approach would be to analyze each of the pieces separately and then combine these inferences at the end,

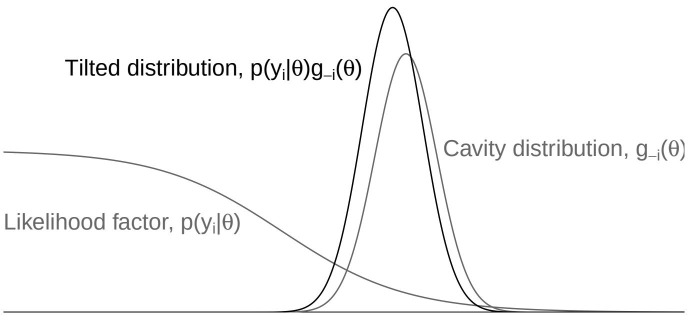

Figure 2: *Example of a step of an EP algorithm in a simple one-dimensional example, illustrating the stability of the computation even when part of the likelihood is far from Gaussian. When performing inference on the likelihood factor $p(y_k|\theta)$, the algorithm uses the cavity distribution $g_{-k}(\theta)$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

to yield a posterior distribution in the overlap. In contrast to data-partitioning algorithms, each step of an EP-based approach combines the likelihood of each partition with the cavity distribution representing the rest of the available information across the other $K-1$ pieces (and the prior). In Figure 1, an EP approach should allow the computations to be concentrated in the area of overlap rather than wasting computation in the outskirts of the individual likelihood distributions.

Figure 2 illustrates the construction of the tilted distribution $g_{\backslash k}(\theta)$ (step 3b of the algorithm) 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 3c) will focus on areas of greater posterior mass.

The remaining conceptual and computational complexity lies in step 3c, the fitting of the updated local approximation $g_k(\theta)$ such that $g_k(\theta)g_{-k}(\theta)$ approximates the tilted distribution $g_{\backslash k}(\theta) = p(y_k|\theta)g_{-k}(\theta)$. This step is in most settings the crux of EP, in terms of computation, accuracy, and stability of the iterations. As such, here we consider a variety of choices for forming tilted approximations, beyond the standard choices in the EP-literature. We call any method of this general iterative approach an *EP-like algorithm.* Thus we can think of the EP-like algorithms described in this paper as a set of rough versions of EP, or perhaps more fruitfully, as a statistical and computational improvement upon partitioning algorithms that analyze each part of the data separately.

3. EP-like algorithm for hierarchical models

The exciting idea for hierarchical models is that the local parameters can be partitioned, and the convergence of the EP-like algorithm only needs to happen on the shared parameters. Suppose a hierarchical model has local parameters $\alpha_1, \dots, \alpha_K$ and shared parameters ϕ. All these can be vectors, with each α_k applying to the model for the data piece y_k, and with ϕ including shared parameters ("fixed effects") of the data model and hyperparameters as well. This structure is displayed in Figure 3.

3.1. An EP-like algorithm on the shared parameters

We will define an EP-like algorithm on the vector of shared parameters, ϕ, so that the kth approximating step, g_k combines the local information at site k with the cavity distribution $g_{-k}(\phi)$ which approximates the other information in the model relevant to this updating. The EP-like algorithm,

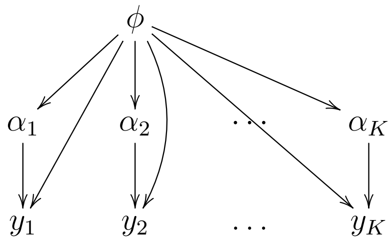

Figure 3: *Model structure for the hierarchical EP-like algorithm. In each step k, inference is based on the local model,* $p(y_k|\alpha_k, \phi)p(\alpha_k|\phi)$*, multiplied by the cavity distribution,* $g_{-k}(\phi)$*. Computation on this tilted posterior gives a distributional approximation on* (α_k, 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 *steps, both the sample size and the number of parameters scale like* $1/K$*.*

taking the steps of the algorithm in the previous section and altering them appropriately, becomes:

1. **Partitioning.** Partition the data y into K pieces, $y_1, \ldots, y_K$, each with its local model $p(y_k|\alpha_k, \phi)p(\alpha_k|\phi)$. The goal is to construct an approximation $g(\phi)$ for the marginal posterior distribution of the shared parameters ϕ. From this, we can also get an approximate joint distribution as described in step 4 below.

2. **Initialization.** Choose initial approximations $g_k(\phi)$ from some restricted family (for example, multivariate normal distributions in ϕ). Define $g(\phi) = p(\phi)\prod_{k=1}^{K} g_k(\phi)$.

3. **EP-like iteration.** For $k = 1, \ldots, K$ (in serial or parallel):

   (a) Compute the cavity distribution, $g_{-k}(\phi) = g(\phi)/g_k(\phi)$.

   (b) Form the *local tilted distribution*, $g_{\backslash k}(\alpha_k, \phi) = p(y_k|\alpha_k, \phi)p(\alpha_k|\phi)g_{-k}(\phi)$, and determine its marginal distribution, $g_{\backslash k}(\phi)$; this can be computed analytically (if $g_{\backslash k}(\alpha_k, \phi)$ is a joint normal distribution) or approximated using the sampled values of ϕ if $g_{\backslash k}(\alpha_k, \phi)$ is computed via simulation.

   (c) Construct an updated site approximation g_k^{new} such that $g_k^{\text{new}}(\phi)g_{-k}(\phi)$ approximates $g_{\backslash k}(\phi)$.

   (d) *If parallel*, set $g_k(\phi)$ to $g_k^{\text{new}}(\phi)$, and a new approximation $g(\phi) = p(\phi)\prod_{k=1}^{K} g_k(\phi)$ will be formed and redistributed after the K site updates. *If serial*, update the global approximation $g(\phi)$ to $g_k^{\text{new}}(\phi)g_{-k}(\phi)$.

4. **Termination.** Repeat step 3 until convergence of the approximate posterior distribution $g(\phi)$. Given this approximation, define an approximate joint distribution $g(\alpha, \phi) = g(\alpha_1, \ldots, \alpha_K, \phi) = g(\phi)\prod_{k=1}^{K} p(y_k|\alpha_k, \phi)p(\alpha_k|\phi)$.

The computational advantage of this algorithm is that the local parameters α are partitioned. For example, suppose we have a model with 100 data points in each of 3000 groups, 2 local parameters per group (a varying slope and intercept) and, say, 20 shared parameters (including fixed effects and hyperparameters). If we then divide the problem into $n = 3000$ pieces, we have reduced a 300,000 × 6020 problem to 3000 parallel 100 × 22 problems. To the extent that computation costs are proportional to sample size multiplied by number of parameters, this is a big win.

3.2. Example: a hierarchical nonlinear regression

We illustrate with an example of a hierarchical regression model from a current research project in astronomy.

Researchers are interested in the relation between fluxes of two different bands of electromagnetic radiation, which are measured as y_i and x_i, respectively, at a large number of locations i. The sky is divided into grid boxes $j = 1, \ldots, J$, and in each grid box j we want to fit a nonlinear regression with likelihood $p(y_j|a_j, \gamma)$, where y_j is the vector of data points in grid box j, a_j is a vector of parameters (for example, regression coefficients) corresponding to grid box j, and γ is some set of shared parameters (for example, regression coefficients, scale parameters, and shape parameters that are constant in the model and do not vary spatially). The local parameter vectors a_j are modeled as independent draws from a multivariate normal distribution, $a_j \sim \mathrm{N}(M, S)$. Finally, for computational reasons the problem is partitioned into $K < J$ pieces, so that each piece contains some number of grid boxes. For example, we might have 10^9 data points and $J = 10^6$ grid boxes, with the problem divided into $K = 10^3$ pieces.

In the notation of section 3.1, the set of shared parameters is $\phi = (\gamma, M, S)$ and, within each piece k, the set of local parameters is $\alpha_k = (a_j)_{\text{for all } j \text{ in piece } k}$. Performing full Bayes (or even computing a Bayesian point estimate) on this model would be costly if the number of data points and the number of grid boxes are large. Our EP formulation breaks down the problem into K smaller problems, each with only $1/K$ as many data points and roughly $1/k$ as many parameters.

We assume a multivariate normal approximate distribution on ϕ—actually, we would transform the parameters to be unconstrained, thus working on the scale of the logarithms of any scale parameters and an appropriately transformed Cholesky decomposition of the covariance matrix S (see Stan Development Team, 2014, section 53.9). We write the approximate distribution as:

$$g(\phi) = \prod_{k=1}^{K} g_k(\phi) = \prod_{k=1}^{K} \mathrm{N}(\phi|\mu_k, \Sigma_k).$$

The μ_k's and Σ_k's are updated during the steps of the EP-like algorithm until convergence.

For simplicity we assume a flat prior on the (unconstrained transformed components of) ϕ and we correspondingly set $g_0(\phi)$ to a constant. (Appendix A describes an algorithm with informative priors.) We start with weak approximations such as setting, for each k, $\mu_k = \vec{0}$ and $\Sigma_k = 10^2 I/K$, so that the initial setting for $g(\phi)$ is normal with mean 0 and scale equal to 10 times the identity matrix.

The EP-like iterations (step 3 of the algorithm in section 3.1) then proceed as follows:

3. For $k = 1, \ldots, K$ (in serial or parallel):

   (a) The cavity distribution is $g_{-k}(\phi) = \prod_{k' \neq k} \mathrm{N}(\phi|\mu_{k'}, \Sigma_{k'}) = \mathrm{N}(\mu_{-k}, \Sigma_{-k})$, where

$$\begin{aligned} \Sigma_{-k}^{-1} &= \sum_{k \not\leq k} \Sigma_{k'}^{-1} \\ \Sigma_{-k}^{-1}\mu_{-k} &= \sum_{k' \neq k} \Sigma_{k'}^{-1}\mu_{k'} \end{aligned}$$

   (b) The tilted distribution is

$$\begin{aligned} g_{\backslash k}(\alpha_k, \phi) &= g_{-k}(\phi) p(\alpha_k, y_k|\phi) \\ &= \mathrm{N}(\phi|\mu_{-k}, \Sigma_{-k}) \prod_{j \text{ in piece k}} \mathrm{N}(a_j|M, S) p(y_j|a_j, \gamma). \end{aligned}$$

We can feed this distribution into Stan—it is simply the likelihood for the data in piece j, the priors for the local parameters a_j within that piece, and a normal prior on the shared parameters determined by the cavity distribution—and then we can directly approximate $g_{\backslash k}(\alpha_k, \phi)$ by a normal distribution (perhaps using a Laplace approximation, perhaps using some quadrature or importance sampling to better locate the mean and variance of this tilted distribution) or else simply take some number of posterior draws. Or a more efficient computation may be possible using importance weighting of earlier sets of simulations at this node, as we discuss in section 4.6. In any case we can get a normal approximation to the marginal tilted distribution of the shared parameters, $g_{\backslash k}(\phi)$. Call this approximation, $\mathrm{N}(\mu_{\backslash k}, \Sigma_{\backslash k})$.

(c) The updated approximate site distribution is $g_k^{\text{new}} = \mathrm{N}(\mu_k^{\text{new}}, \Sigma_k^{\text{new}})$, set so $g_k^{\text{new}}(\phi)g_{-k}(\phi)$ approximates $g_{\backslash k}(\phi)$. It would be most natural to do this by matching moments, but it is possible that the differencing step could result in non-positive-definite covariance matrix Σ_k^{new}, so some more complicated step might be needed, as discussed in section 4.7.

(d) We can then perform the appropriate parallel or serial update.

As this example illustrates, the core computation is the inference on the local tilted distribution in step b, and this is where we are taking advantage of the partitioning, both in reduction of the data and in reduced dimensionality of the computation, as all that is required is inference on the shared parameters ϕ and the local parameters α_k, with the other $K-1$ sets of local parameters being irrelevant in this step.

3.3. Posterior inference for the approximate joint distribution

Once convergence has been reached for the approximate distribution $g(\phi)$, we approximate the joint posterior distribution by $g(\alpha_1, \ldots, \alpha_K, \phi) = g(\phi) \prod_{k=1}^{K} p(y_k|\alpha_k, \phi) p(\alpha_k|\phi)$. We can work with this expression in one of two ways, making use of the ability to simulate the model in Stan (or otherwise) in each note.

First, if all that is required are the separate (marginal) posterior distributions for each α_k vector, we can take these directly from the simulations or approximations performed in step 3b of the algorithm, using the last iteration which can be assumed to be at approximate convergence. This will give simulations of α_k or an approximation to $p(\alpha_k, \phi|y)$.

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

4. Open issues in implementation

4.1. Algorithmic considerations

Implementing an EP-like algorithm involves several decisions:

- Algorithmic blocking. This includes parallelization and partitioning of the likelihood. In the present paper we a assume a model with independent data points conditional on its parameters so the likelihood can be factored. The number of subsets K will be driven by computational considerations. If K is too high, the Gaussian approximation will not be so accurate. However, if K is low, then the computational gains will be small. For large problems it could make sense to choose K iteratively, for example setting to a high value and then decreasing it if the approximation seems too poor. Due to the structure of modern computer memory, the computation using small blocks may get additional speed-up if the most of the memory accesses can be made using fast but small cache memory.

- The parametric form of the approximating distributions $g_k(\theta)$. We will use the multivariate normal. For simplicity we will also assume that the prior distribution $p_0(\theta)$ also is multivariate normal, as is the case in many practical applications, sometimes after proper reparameterization (such as with constrained parameter spaces). 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.

- The starting distributions g_k. We will use a broad but proper distribution factored into K equal parts, for example setting each $g_k(\theta) = \mathrm{N}(0, \frac{1}{n}A^2 I)$, where A is some large value (for example, if the elements of θ are roughly scaled to be of order 1, we might set $A = 10$).

- The algorithm to perform inference on the tilted distribution. We will consider three options: deterministic mode-based approximation (in section 4.3), deterministic improvements of mode-based approximations (in section 4.4) and Bayesian simulation (in section 4.5).

- The approximation to the tilted distribution. For mode-based approximations, we can calculate the first two moments of the fitted normal or other approximate density; see section 4.4. Or if the tilted distribution is computed by Monte Carlo methods we can compute the first two moments of the simulation, possibly with some regularization if θ is of high dimension; see section 4.5.

- The division by the cavity distribution in step 3c can yield a non-positive-definite variance matrix which would correspond to a meaningless update for g_k. We can stabilize this by setting negative eigenvalues to small positive values or perhaps using some more clever method. We discuss this in section 4.7.

In a hierarchical setting, the model can be fit using the nested EP approach (Riihimäki et al., 2013). Assuming that the likelihood term associated with each data partition is approximated with a local Gaussian site function $g_k(\alpha_k, \phi)$, the marginal approximations for ϕ and all α_k could be estimated without forming the potentially high-dimensional joint approximation of all unknowns. At each iteration, first the K local site approximations could be updated in parallel with a fixed marginal approximation $g(\phi)$. Then $g(\phi)$ could be refreshed by marginalizing over α_k separately using the new site approximations and combining the marginalized site approximations. The parallel EP approximations for each data partition correspond to the inner EP approximations, and inference for $g(\phi)$ corresponds to the outer EP.

4.2. Approximating the tilted distribution

In standard EP, the tilted distribution approximation in step 3c is done by moment matching problem, which when using the multivariate normal family implies that: 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 intractable tilted distribution $g_{\backslash k}(\theta)$. When used for Gaussian processes, this approach has the particular advantage that the tilted distribution $g_{\backslash k}(\theta)$ can typically be set up as a univariate distribution over only a single dimension in θ. This implicit dimension reduction implies that the tilted distribution approximation can be performed analytically (e.g., Minka, 2001) or relatively quickly using one-dimensional quadrature (e.g., Zoeter and Heskes, 2005). In higher dimensions, quadrature gets computationally more expensive or, with reduced number of evaluation points, the accuracy of the moment computations gets worse. Seeger (2004) estimated the tilted moments in multiclass classification using multidimensional quadratures. Without the possibility of dimension reduction in the more general case, there is no easy way to compute or approximate the integrals to compute the required moments over $\theta \in \mathbb{R}^k$. Accordingly, black-box EP would seem to be impossible.

To move towards a black-box EP-like algorithm, we can change this moment matching choice to instead match the mode or use numerical simulations. The resulting EP-like algorithms critically preserve the essential idea that the local pieces of data are analyzed at each step in the context of a full posterior approximation. These alternative choices for a tilted approximation generally make EP less stable, and we consider methods for fast and accurate approximations and stabilizing computations for EP updates.

Smola et al. (2004) proposed Laplace propagation, where moment-matching is replaced with a Laplace approximation, so that the tilted mean is replaced with the mode of the tilted distribution and the tilted covariance with the inverse Hessian of the log density at the tilted mode. 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. Riihimaki et al. (2013) presented a nested EP method, where moments of the multivariate tilted distribution are also estimated using EP. For certain model types the inner EP can be computed efficiently.

In this paper, we also consider MCMC computation of the moments, which we suspect will give inaccurate moment estimates, but may work better than, or as a supplement to, Laplace approximation for skewed distributions. We also propose an importance sampling scheme to allow stable estimates based on only a moderate number of simulation draws.

When the moment computations are not accurate, EP may have stability issues, as discussed by 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. Fractional EP (Minka, 2004) is an extension of EP which can be used 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). Fractional EP can be viewed as a method for minimizing of the α-divergence, with $\alpha = 1$ corresponding to Kullback-Leibler divergence used in EP, $\alpha = 0$ corresponding to the reverse Kullback-Leibler divergence usually used in variational Bayes, and $\alpha = 0.5$ corresponding to Hellinger distance. Ideas of fractional EP might help to stabilize EP-like algorithms that use approximative moments, as α-divergence with $\alpha < 1$ is less sensitive to errors in tails of the approximation.

Section 4.1 details these and other considerations for tilted approximations in our EP-like algorithm framework. While the tilted approximation is the key step in any EP-like algorithm, there are two other canonical issues that must be considered in any EP-like approach: lack of convergence guarantees and the possible computational instability of the iterations themselves. Section 4.1 also considers approaches to handle these instabilities.

4.3. Normal approximation based on deterministic computations

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

The most basic approximation is obtained by, at each step, setting g^{new} to be the (multivariate) normal distribution centered at the mode of $g_{\backslash k}(\theta)$, with covariance matrix equal to the inverse of the negative second derivative matrix of $\log g_{\backslash k}$ at the mode. This corresponds to the Laplace approximation (Smola et al., 2004). We assume any parameters with restricted range have been transformed to unrestricted spaces (in Stan, this is done via logarithms of all-positive parameters, logits of interval-constrained parameters, and special transforms for simplex constraints and covariance matrices; see Stan Development Team, 2014, section 53.9), so the gradient of $g_{\backslash k}(\theta)$ will necessarily be zero at any mode.

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

4.4. Split-normal and split-t approximations

After computing the mode and curvature at the mode, 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 aspects of asymmetry and long tails in the posterior distribution. Possible approximate families include the multivariate split-normal (Geweke, 1989, Villani and Larsson, 2006), split-t, or wedge-gamma (Gelman et al., 2014) distributions.

We are *not* talking about changing the family of approximate distributions g—we still would keep these as multivariate normal. Rather, we would use an adaptively-constructed parametric approximation, possibly further improved by importance sampling (Geweke, 1989) or CCD integration (Rue et al., 2009) to get a better approximation to the mean and covariance matrix of the tilted distribution to use in constructing g_k in step 3c of the algorithm.

4.5. Estimating moments using simulation

A different approach is at each step to use simulations (for example, Hamiltonian Monte Carlo using Stan) to approximate the tilted distribution and then use these to set the mean and covariance of the approximation in step 3c. As above, the advantage of the EP-like algorithm here is that the computation only uses a fraction $1/K$ of the data, along with a simple multivariate normal prior that comes from the cavity distribution. Similar to other inference methods such as stochastic variational inference (Hoffman et al., 2013) which take steps based on stochastic estimates, EP update steps need unbiased estimates. When working with the normal approximation, we then need unbiased estimates of the mean and precision in ((1)). The variance of the estimates can be reduced while preserving unbiasedness by using control variates.

4.6. Reusing simulations with importance weighting

In serial or parallel EP, samples from previous iterations can be reused as starting points for either Markov chains or in importance sampling. We discuss briefly the latter.

Assume we have obtained at iteration t for node k, a set of posterior simulation draws $\theta^s_{t,k}$, $s = 1, \ldots, S_{t,k}$ from the tilted distribution $g^t_{\backslash k}$, possibly with weights $w^s_{t,k}$; take $w^s_{t,k} \equiv 1$ for an unweighed sample.

To progress to node $k+1$, reweight these simulations as: $w^s_{t,k+1} = w^s_{t,k} g^t_{\backslash(k+1)}(\theta^s_{t,k})/g_{\backslash k}(\theta^s_{t,k})$. If the effective sample size (ESS) of the new weights,

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

is large enough, keep this sample, $\theta^s_{t,k+1} = \theta^s_{t,k}$. Otherwise throw it away, simulate new $\theta^s_{t+1,k}$'s from $g^t_{\backslash k+1}$, 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). Elaborations could be considered. Instead of throwing away a sample with too low an ESS, one could move these through several MCMC steps, in the spirit of sequential Monte Carlo (Del Moral et al., 2006).

Another approach, which can be used in serial or parallel EP, is to use adaptive multiple importance sampling (Cornuet et al., 2012), which would make it possible to recycle the simulations from previous iterations.

Even the basic strategy should provide important savings when EP is close to convergence. Then changes in the tilted distribution should become small and as 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.

4.7. Keeping the covariance matrix positive definite

When working with the normal approximation, step 3c of the EP-like algorithm can be conveniently written in terms of the natural parameters of the exponential family:

$$\begin{aligned} (\Sigma^{\text{new}}_k)^{-1}\mu^{\text{new}}_k &= (\Sigma^{\text{new}})^{-1}\mu^{\text{new}} - \Sigma^{-1}_{-k}\mu_{-k} \\ (\Sigma^{\text{new}}_k)^{-1} &= (\Sigma^{\text{new}})^{-1} - \Sigma^{-1}_{-k}, \end{aligned} \tag{1}$$

where μ^{new} and Σ^{new} are the approximate mean and covariance matrix of the tilted distribution, and these are being used to determine μ^{new}_k and Σ^{new}_k, the mean and variance of the updated site distribution g_k. The difficulty is that the difference between two positive-definite matrices is not itself necessarily positive definite. There are various tricks in the literature to handle this problem. One idea is to first perform the subtraction in the second line above, then do an eigendecomposition, keeping the eigenvectors of $(\Sigma^{\text{new}}_k)^{-1}$ but taking any negative eigenvalues and replacing them with small positive numbers as in the SoftAbs map of Betancourt (2013). It is not clear whether some similar adjustment would be needed for the updating of $\Sigma^{-1}_{-k}\mu_{-k}$ or whether the top line above would work as written.

Is it possible to take into account the positive-definiteness restriction in a more principled manner? If our measure of the precision matrix of the tilted distribution is noisy (due to Monte Carlo error), it would make sense to include this noise in our inference and estimate μ^{new}_k and Σ^{new}_k statistically via a measurement-error model. The noisy precision of the tilted distribution is the sum of the cavity precision, the precision of the pseudo-observation, and the noise term. Can we infer from this better estimate of precision of the pseudo-observation? This itself would be a small Bayesian inference problem but of low enough dimensionality that it should not represent a large part of the computation cost compared to the other steps of the algorithm.

An simple and effective alternative is to damp each step so that the resulting cavity covariance remains positive definite after each update. This does not change the fixed points of the EP algorithm, and adjusts only the step sizes during the iterations. If the updates are done in parallel fashion, it is possible to derive a limit on the damping factor (or step size) so that the cavity covariances at all sites remain positive definite. The downside is that this requires determining one eigendecomposition at each site at each parallel update. Based on our experiments, this takes negligible time compared to the tilted distribution moment computations.

4.8. The computational opportunity of parallel EP-like algorithms

We have claimed that EP-like 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 (section 2) with a multivariate normal approximating family. We assume that we have $K+1$ parallel processing units: one central processor that maintains the global posterior approximation $g(\theta)$ and K worker units on which inference can be computed on each of the K factors of the likelihood. Furthermore, we assume a network transmission cost of c per parameter. Let N be the number of data points and let D be the number of parameters, that is, the length of the vector θ. Finally, we define $h(n, d)$ as the computational cost of approximating the tilted distribution (a task which may, for example, be performed by running MCMC to perform simulations) with n data points and d parameters.

Each step of the algorithm then incurs the following costs:

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

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

3. **EP-like 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, 2009).

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

   (b) Forming the tilted distribution. This conceptual step bears no cost.

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

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

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

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

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

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

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

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

$$O\left(h(N, D)\right).$$

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

5. Experiments

5.1. Implementation in Stan

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

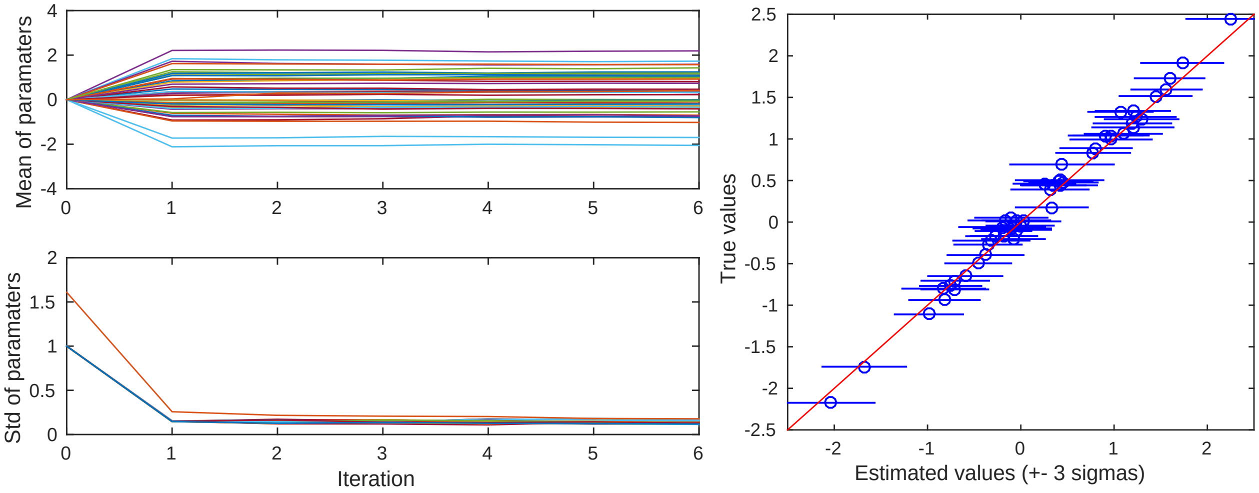

Figure 4: *For simulated logistic regression with 50 parameters and $N = 2500$ data points in $J = 50$ groups divided into $K = 50$ pieces: (a) Left plots show the convergence of the EP approximation for the shared paramaters; (b) Right plot compares the EP inferences for the shared parameters to their true values. Convergence of the algorithm is nearly instantaneous, illustrating the effectiveness of dispersed computation in this example.*

We use Stan to compute step 3a of the EP-like algorithm, the inference for the tilted distribution for each process. Currently we perform the other steps in Matlab, Python, or R (code will be made available at `https://github.com/gelman/ep-stan`). In parallel EP, we pass the normal approximations g_k back and forth between the master node and the K separate nodes.

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

5.2. Hierarchical logistic regression

We test the algorithm with a simple hierarchical logistic regression,

$$\begin{aligned} y_i &\sim \text{Bernoulli}(\alpha_j + \beta x_{ij}) \\ \alpha_j &\sim \text{N}(0, \tau^2), \end{aligned}$$

where β is a vector of common coefficients and each group has own intercept α_j with a hierarchical prior. The data were simulated using $x_{ij} \sim \text{N}(0, 1)$, $\alpha_j \sim \text{N}(0, 2^2)$ and $\beta \sim \text{N}(0, 1)$.

We first perform a simulation experiment with 50 regression predictors, the number of groups $J = 50$, the number of data in each group $N_j = 50$, and thus a total sample size of $N = 2500$. For simplicity we partition the data as one piece per group; that is, $K = J$. The Stan runs had 50 warmup and 200 iterations per chain.

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 problematic: with only 50 data points per group, each of the local posterior distributions would be very wide (as sketched in Figure 1). The EP-like framework, in which at each step the cavity distribution is used as a prior, keeps computations more stable and focused.

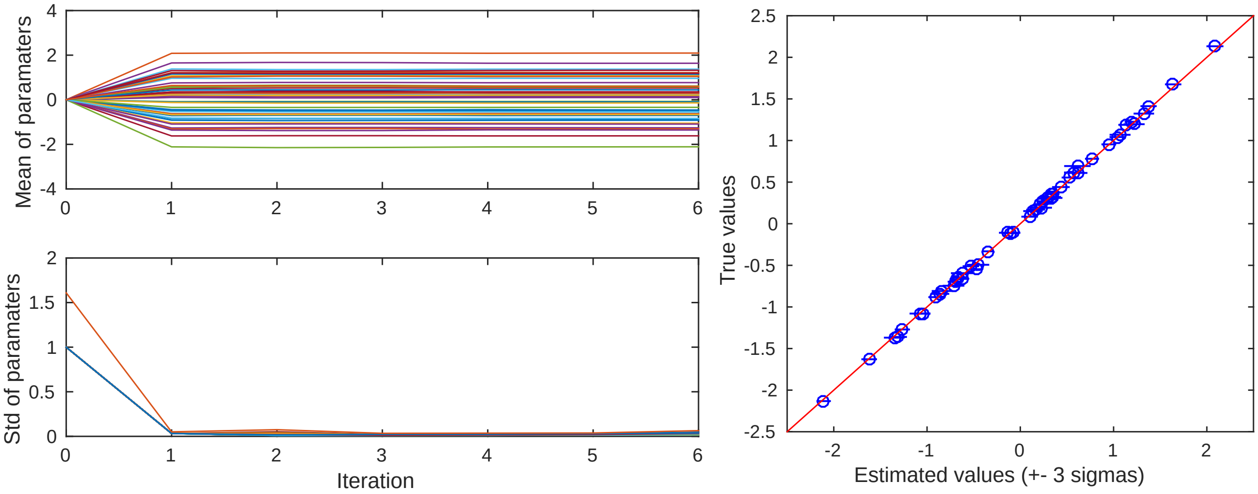

Figure 5: *For simulated logistic regression with 50 parameters and* $N = 50{,}000$ *data points in* $J = 1000$ *groups divided into* $K = 50$ *pieces: (a) Left plots show the convergence of the EP approximation for the shared paramaters; (b) Right plot compares the EP inferences for the shared parameters to their true values. With such a large dataset the normal approximation is no problem; we implemented a simulation of this size to begin to check the scalability of the algorithm.*

Figure 4 shows the results: convergene is essentially immediate, and the algorithm ran fast. We have not yet performed runtime comparisons because our implementation has many places where we can increase its speed. But it is reassuring that the computations work well in this very simple and direct MCMC implementation of our algorithm.

In addition to checking the algorithm's convergence (left plots in Figure 4), we also check the coverage of posterior inferences from the simulations (right plot). If we were performing full Bayesian inference this coverage check would not be necessary (except possibly for catching bugs), but because the EP-like algorithm is approximate, it makes sense to check coverage in this fake-data simulation. As the graph shows, the coverage is fine, with all 50 estimates within 3 standard errors of the true values, the vast majority being within 2 standard errors, and about two-thirds being within 1 standard error of the truth. For a fuller evaluation of the algorithm it would be a good idea to study this sort of coverage more thoroughly; here we just wanted to get a rough sense that the results made sense and that the approximating distribution was not clearly too narrow or two broad.

Figure 5 shows results for a new set of simulations, this time again with 50 regression predictors and $K = 50$ partitions, but this time with $J = 1000$ groups and $N_j = 50$ observations per group, so that the data are partitioned via cluster sampling into 50 partitions with 20 groups and 1000 data points in each partition. By increasing the size of the problem, we are improving the normal approximation but we are beginning to test the scalability of the algorithm. We plan to do more systematic scalability testing once we have increased the efficiency of some of our computational steps.

The experiments show that the algorithm works and is stable even with the simplest moment computation using MCMC. All the above experiments were made on a desktop computer without any actual parallelization. Additional modifications suggested in section 4 should further speed up the moment computations and the convergence.

6. Discussion

In this paper we have presented a framework for using the EP idea of cavity and tilted distributions to perform Bayesian inference on partitioned datasets. Particularly in the case of hierarchical models, this structure can allow efficient dispersed computation for big models fit to big data.

6.1. 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 to each approximating site g_k a constant Z_k, and write the global approximation as:

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

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

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

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

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

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

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

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

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

6.2. 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, $g_{\backslash k_1, k_2} = g \frac{p_{k1} p_{k2}}{g_{k1} g_{k2}}$, which we might call the tilted^2 distribution;
- For any k_1, k_2, k_3, $g_{\backslash k_1, k_2, k+3} = g \frac{p_{k1} p_{k2} p_{k3}}{g_{k1} g_{k2} g_{k3}}$, the tilted^3 distribution;
- $\ldots$

- $p = \prod_{k=0}^{K} p_k$, the target distribution, which is also the tilted^K distribution.

Instead of independent groups, also tree structures could be used (Minka, 2001). As presented, the goal of an EP-like algorithm is to determine g. But at each step we get simulations from all the $g_{\backslash k}$'s, so we could try to combine these inferences in some way (for example, following the ideas of Wang and Dunson, 2013). Even something as simple as mixing the simulation draws from the tilted distribution could be a reasonable improvement on the EP approximation. One could then go further, for example at convergence computing simulations from some of the tilted^2 distributions.

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

6.3. Connections to other approaches

The EP-like algorithm 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 $g = p_0 \prod_{k=1}^{K} g_k$, but 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).

Data partitioning is an extremely active research area right now, with several different black-box algorithms being proposed by various research groups. We are sure that different methods will be more effective in different problems. The present paper has two roles: we are presenting a particular black-box algorithm for distributional approximation, and we are suggesting a general approach to approaching data partitioning problems. We anticipate that great progress could be made by using EP ideas to regularize existing algorithms.

References

Ahn, S., Korattikara, A., and Welling, M. (2012). Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning.*

Barthelmé, S., and Chopin, N. (2014). Expectation propagation for likelihood-free inference. *Journal of the American Statistical Association* **109**, 315–333.

Betancourt, M. (2013). A general metric for Riemannian manifold Hamiltonian Monte Carlo. `http://arxiv.org/abs/1212.4693`

Betancourt, M. (2014). Adiabatic Monte Carlo. `http://arxiv.org/pdf/1405.3489.pdf`

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

Cornuet, J. M., Marin, J. M., Mira, A., and Robert, C. P. (2012). Adaptive multiple importance sampling. *Scandinavian Journal of Statistics* **39**, 798–812.

Cseke, B., and Heskes, T. (2011). Approximate marginals in latent Gaussian models. *Journal of Machine Learning Research* **12**, 417–454.

Cunningham, J. P., Hennig, P., and Lacoste-Julien, S. (2011). Gaussian probabilities and expectation propagation. `http://arxiv.org/abs/1111.6832`

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society B* **68**, 411–436.

Gelman, A., Bois, F. Y., and Jiang, J. (1996). Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association* **91**, 1400–1412.

Gelman, A., Carpenter, B., Betancourt, M., Brubaker, M., and Vehtari, A. (2014). Computationally efficient maximum likelihood, penalized maximum likelihood, and hierarchical modeling using Stan. Technical report, Department of Statistics, Columbia University.

Gelman, A., Jakulin, A., Pittau, M. G., and Su, Y. S. (2008). A weakly informative default prior distribution for logistic and other regression models. *Annals of Applied Statistics* **2**, 1360–1383.

Gershman, S., Hoffman, M., and Blei, D. (2012). Nonparametric variational inference. In *Proceedings of the 29th International Conference on Machine Learning.*

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

Girolami, M., and Zhong, M. (2007). Data integration for classification problems employing Gaussian process priors. In *Advances in Neural Information Processing Systems 19*, ed. B. Scholkopf, J. Platt, and T. Hoffman, 465–472.

Hoffman, M., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1), 1303-1347.

Huang, Z., and Gelman, A. (2005). Sampling for Bayesian computation with large datasets. Technical report, Department of Statistics, Columbia University.

Jylänki, P., Vanhatalo, J., and Vehtari, A. (2011). Robust Gaussian process regression with a Student-t likelihood. *Journal of Machine Learning Research* **12**, 3227-3257.

Korattikara, A., Chen, Y., and Welling, M. (2013). Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of the 31st International Conference on Machine Learning.*

Minka, T. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, ed. J. Breese and D. Koller, 362–369.

Minka, T. (2004). Power EP. Technical report, Microsoft Research, Cambridge.

Minka, T. (2005). Divergence measures and message passing. Technical report, Microsoft Research, Cambridge.

Minkser, S., Srivastava, S., Lin, L., and Dunson, D. B. (2014). Robust and scalable Bayes via a median of subset posterior measures. Technical report, Department of Statistical Science, Duke University. `http://arxiv.org/pdf/1403.2660.pdf`

Neiswanger, W., Wang, C., and Xing, E. (2013). Asymptotically exact, embarrassingly parallel MCMC. `arXiv:1311.4780`.

Rasmussen, C. E., and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning.* Cambridge, Mass.: MIT Press.

Riihimaki, J., Jylänki, P., and Vehtari, A. (2013). Nested expectation propagation for Gaussian process classification with a multinomial probit likelihood. *Journal of Machine Learning Research* **14**, 75–109.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society B* **71**, 319–392.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2013). Bayes and big data: The consensus Monte Carlo algorithm. In *Bayes 250.*

http://research.google.com/pubs/pub41849.html

Seeger, M. (2005). Expectation propagation for exponential families. Technical report, Max Planck Institute for Biological Cybernetics, Tubingen, Germany.

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

Seeger, M., and Jordan, M. (2004). Sparse Gaussian process classification with multiple classes. Technical report, University of California, Berkeley.

Smola, A., Vishwanathan, V., and Eskin, E. (2004). Laplace propagation. In *Advances in Neural Information Processing Systems 16*, ed. S. Thrun, L. Saul, and B. Scholkopf.

Stan Development Team (2014). Stan modeling language: User's gude and reference manual, version 2.5.0. http://mc-stan.org/

van Gerven, M., Cseke, B., Oostenveld, R., and Heskes, T. (2009). Bayesian source localization with the multivariate Laplace prior. In *Advances in Neural Information Processing Systems 22*, ed. Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, 1901–1909.

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., and Vehtari, A. (2013). GPstuff: Bayesian modeling with Gaussian processes. *Journal of Machine Learning Research* **14**, 1005–1009.

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

Wang, C., and Blei, D. M. (2013). Variational inference in nonconjugate models. *Journal of Machine Learning Research* **14**, 899–925.

Wang, X. (2014). Parallel MCMC via Weirstrass sampler. Xi'an's Og, 3 Jan. http://xianblog.wordpress.com/2014/01/03/parallel-mcmc-via-weirstrass-sampler-a-reply-by-xiangyu-wang/

Wang, X., and Dunson, D. B. (2013). Parallel MCMC via Weierstrass sampler. http://arxiv.org/pdf/1312.4605v1.pdf

Zoeter, O., and Heskes, T. (2005). Gaussian quadrature based expectation propagation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, ed. Z. Ghahramani and R. Cowell, 445–452.

A. Distributed parallel algorithms

We have pitched this article at a fairly abstract level because we are interested in the general idea of using EP-like algorithms to perform parallel Bayesian computation using data splitting. Ultimately, though, these algorithms are motivated by speed and scalability concerns, so we need efficient implementation. In this appendix we go through some of the steps that we went through to efficiently implement the simulations described in section 5.

A.1. Distributed parallel EP-like algorithm

In this appendix we give a practical algorithm description that is suitable for implementing the general EP-like algorithms of sections 2 and 3 in a numerically stable manner. We assume that the posterior distribution,

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

is approximated by

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

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

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

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

$$Q = \Sigma^{-1} = \sum_{k=1}^{K} Q_k + Q_0 \quad \text{and} \quad r = \Sigma^{-1}\mu = \sum_{k=1}^{K} r_k + r_0. \tag{3}$$

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

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

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

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

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

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

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

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

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

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

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

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

   which can be efficiently sampled and differentiated using

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

   Key properties of the different approximation methods:

   - MCMC: It is easy to compute $\mu_{\backslash k}$ and $\Sigma_{\backslash k}$ from a set of samples, and $\Sigma_{\backslash k}$ should be symmetric and positive definite if enough samples are used. However, computing the precision matrix $Q_{\backslash k} = \Sigma_{\backslash k}^{-1}$ requires a $O(d^3)$ Cholesky or eigenvalue decomposition. Could this be done simultaneously within the SoftAbs stabilization step? In addition, there should be literature on estimating (sparse) precision matrices from samples, which could be beneficial with large k.
   - Laplace's method: Gradient-based methods can be used to determine the mode of the tilted distribution efficiently. Once a local mode $\hat{\theta}$ is found, the natural parameters can be computed as

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

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

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

$$\begin{aligned}\Delta Q_k &= Q_k^{\text{new}} - Q_k = \eta^{-1}(Q_{\setminus k} - Q_{-k}) - Q_k \\ \Delta r_k &= r_k^{\text{new}} - r_k = \eta^{-1}(r_{\setminus k} - r_{-k}) - r_k,\end{aligned}$$

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

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

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

Advantages of the approach include:

- The central node does not need to compute $O(d^3)$ matrix decompositions in step 2. It only needs to sum together the natural site parameters to obtain the posterior approximation in exponential form, and pass this to the individual computing nodes that can make the subtractions to form the cavity parameters.

- The tilted moments can be determined by sampling directly from the unnormalized tilted distributions or by using the Laplace's method. This requires only cheap function and gradient evaluations and can be applied to 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 $g(\theta)$. This sample-based approximation could capture also potential skewness in $p(\theta|y)$ because it resembles the EP-based marginal improvements described by Cseke and Heskes (2011).

Limitations of the approach include:

- The tilted covariance matrices can be easily computed from samples, but these have to be inverted to obtain the site precision matrices. This inversion could be done by each node after sampling, but if there are more efficient ways to approximate (sparse) precision matrices directly from samples, this could be a potential scheme even for large number of unknowns? The algorithm would not require $O(d^3)$ matrix decompositions and all the required posterior marginals could be summarized using samples.

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

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

A.2. Efficient algorithms when dimension reduction is possible

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

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

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

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

and we approximate it by

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Save $c_k = \Psi(r_{-k}, Q_{-k}) - \Psi(V_k^{-1} m_k, V_k^{-1})$ for computing the marginal likelihood as described in section 6.1.

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

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

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

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

where $Z_{\backslash k} = \int p(y_k | U' \theta)^{\eta} \mathrm{N}(z_k | m_{-k}, V_{-k}) dz_k$. Save $Z_{\backslash k}$ for computing the marginal likelihood as described in section 6.1.

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

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

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

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

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

Advantages of the approach include:

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

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

Limitations of the approach include:

- The model terms need to depend on low-dimensional transformations of the unknowns $z_k = U_k \theta$. For example generalized linear models and Gaussian processes fall in to this category.

- Different types of model or likelihood terms require specific implementations. For example, probit and finite Gaussian mixture likelihoods can be handled analytically whereas Poisson and student-t likelihoods require quadratures. For a black-box implementation we might prefer to use numerical quadrature for all these problems.

- The central node needs to compute the global posterior covariance at step 2, which scales as $O(d^3)$ and can be tedious with a large number of unknowns. Independence assumptions or multilevel designs as proposed in section 3 can be used to improve the scaling.

A.3. Parallel EP implementation for hierarchical models when approximations are formed also for the local parameters

In this section we describe a distributed EP algorithm that uses the hierarchical model structure from section 3 for efficient parallel computations in inference problems where a Gaussian approximation is required also for the local parameters. Such cases arise, for example, when certain data pieces share some of the local parameters but we do not wish to form the potentially high dimensional joint approximation of ϕ and all the local parameters.

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

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

by

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

where the site approximations and the prior terms are written using (2), and Z_0 is the normalization term of the joint prior $p(\alpha, \phi)$. To derive the EP updates for the hierarchical model, we divide the site location vector r_k and the site precision matrix Q_k to blocks corresponding to α_k and ϕ as

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

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

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

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

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

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

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

where the kth cavity distribution is defined as

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

with natural parameters

$$\begin{aligned} b_{-k} &= \sum_{j \neq i} \left(b_j - C_j' (A_j + A_0)^{-1} (a_j + a_0) \right) + b_0 = b - \left(b_k - C_k' (A_k + A_0)^{-1} (a_k + a_0) \right) \\ B_{-k} &= \sum_{j \neq i} \left(B_j - C_j' (A_j + A_0)^{-1} C_j \right) + B_0 = B - \left(B_k - C_k' (A_k + A_0)^{-1} C_k \right). \end{aligned} \tag{9}$$

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

$$\begin{aligned}\Sigma_{\alpha_k} &= \left(A_0 + A_k - C_k(B_{-k} + B_k)^{-1}C_k'\right)^{-1} \\ \mu_{\alpha_k} &= \Sigma_{\alpha_k}\left(a_0 + a_k - C_k(B_{-k} + B_k)^{-1}(b_{-k} + b_k)\right).\end{aligned} \tag{10}$$

The marginal approximations $g(\phi)$ and $\{g(\alpha_k)\}_{k=1}^K$ can be computed efficiently without actually forming the potentially high-dimensional joint approximation $g(\alpha_1, \ldots, \alpha_K, \phi)$. After convergence, we can summarize the coefficients and compute the predictions for each group $k = 1, \ldots, K$ using the marginal distributions (7) and (10).

Approximations (7) and (10) can be determined by first initializing the site parameters and the parameter updates to zero, that is $(a_k = 0, b_k = 0, A_k = 0, B_k = 0, C_k = 0)_{k=1}^K$ and $(\Delta a_k = 0, \Delta b_k = 0, \Delta A_k = 0, \Delta B_k = 0, \Delta C_k = 0)_{k=1}^K$, and then iterating the following steps until convergence:

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

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

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

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

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

$$\begin{aligned} b^{\text{new}} &= (\Sigma_\phi^{\text{new}})^{-1}\mu_\phi^{\text{new}} = \sum_{k=1}^K \tilde{b}_k + b_0 \\ B^{\text{new}} &= (\Sigma_\phi^{\text{new}})^{-1} = \sum_{k=1}^K \tilde{B}_k + B_0. \end{aligned} \tag{11}$$

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

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

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

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

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

   Otherwise, accept the new state by setting $b = b^{\text{new}}$, $B = B^{\text{new}}$, and $(a_k = a_k^{\text{new}}, A_k = A_k^{\text{new}}, b_k = b_k^{\text{new}}, B_k = B_k^{\text{new}}, C_k = C_k^{\text{new}})_{k=1}^K$ and continue to the next step. Save the normalization terms $\log Z_{-k}$ for computing the marginal likelihood as described in section 6.1.

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

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

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

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

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

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

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

   Save the following quantity for computing the marginal likelihood as described in section 6.1:

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

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

A.4. Determining tilted moments using inner EP approximations when dimension reduction is possible

We can use an inner EP algorithm to determine the natural parameters $r_{\backslash k}$ and $Q_{\backslash k}$ of the tilted distributions $g_{\backslash k}(\alpha_k, \phi)$ in step 5a of the hierarchical EP algorithm in Appendix A.3, if the likelihood terms related to each data piece can be factored into simple terms that depend only on low-dimensional linearly transformed random variables $z_{k,j} = U_{k,j}(\alpha_k, \phi)$, that is,

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

where n_k is the number of observations in batch k. The EP algorithm description with dimension reduction from section $A.2$ can be readily applied for this purpose. Since the tilted moment approximations in step 5a of the hierarchical algorithm are already run in parallel at the different computing nodes, the parallelization steps in section $A.2$ can be excluded when used to from the inner EP approximations.

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

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

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

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

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

With this representation, we can interpret the hierarchical EP algorithm with K inner EP approximations also as a single algorithm with site parameters $\tilde{r}_{k,j}$ and $\tilde{Q}_{k,j}$ that are updated in parallel fashion for K groups of site terms. After each successful update at step 4 we store only parameters $\tilde{r}_{k,j}$ and $\tilde{Q}_{k,j}$, and we can also equivalently apply damping directly to these parameters at step 1. In fact, it is more efficient to initialize each tilted moment approximation at step 5a to the inner EP parameters from the previous iteration instead of starting from a zero initialization. This framework is similar to the nested EP algorithm for the multinomial probit model described by Riihimäki et al. (2013). However, if applied to the potentially high-dimensional hierarchical setting, the computationally benefits become more evident compared with the GP classification case studied in that paper.